PROPOSED INFECTION MODEL FOR THE REVIVAL OF MASS PRODUCTION OF *KAPENG BARAKO* WITH AIDE OF *BACILLUS SUBTILIS*

by

Sean Matthew S. Agacer

A special problem submitted to the
Department of Mathematics and Computer Science
College of Science
The University of the Philippines
Baguio, Baguio City

as partial fulfillment of the
requirements for the degree of
Bachelor of Science in Computer Science

May 2025

This is to certify that this Special Problem entitled **" Proposed Infection Model For The Revival Of Mass Production Of *Kapeng Barako* With Aide Of *Bacillus Subtilis*"**, prepared and submitted by **Sean Matthew S. Agacer** to fulfill part of the requirements for the degree of **Bachelor of Science in Computer Science** , was successfully defended and approved on May 2025.

LEE JAVELLANA
Special Problem Adviser

The Department of Mathematics and Computer Science endorses the acceptance of this Special Problem as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science .

ANDREI A. DOMOGO, PH.D.
Chair
Department of Mathematics and
Computer Science

Table of Contents

Acknowledgments

I would like to express my gratitude to the people who are dear to my heart:

First and foremost, I would like to express my deepest gratitude to my mother - Annaly S. Agacer, whose unwavering support, love, and encouragement have been the foundation of my academic journey. Her strength and sacrifices have inspired me every step of the way.

To my brother - Stephen Paul S. Agacer , thank you for always believing in me and for being a constant source of motivation and laughter during challenging times. Your presence has made this journey more bearable and meaningful.

I am also sincerely grateful to my uncle Nandy, whose guidance and wisdom have helped shape my perspective and provided valuable insights throughout this research.

To my Lolo Boy and Lola Susan, thank you for your unconditional love and for instilling in me the values of perseverance and curiosity. Your stories and support have always reminded me of the importance of education and hard work.

I would like to extend my heartfelt thanks to my adviser, Lee Javellana, for his invaluable mentorship, patience, and constructive feedback. Your guidance has been instrumental in the completion of this research.

Lastly, to my friends—thank you for your camaraderie, encouragement, and for being there through the highs and lows. Your support has meant the world to me.

This work is dedicated to all who believed in me even when I doubted myself.

Abstract

Proposed Infection Model For The Revival Of Mass Production Of *Kapeng Barako* With Aide Of *Bacillus Subtilis*

Sean Matthew S. Agacer
University of the Philippines, 2025

Adviser:
Lee Javellana

Kapeng Barako - a variety of *Coffea Liberica*in the Philippines, was severely lost due to Coffee Rust disease and is now classified as a protected and conserved. This study proposed a model of *Kapeng Barako* plantation along with the Coffee Rust Disease and *Bacillus subtilis* as the fungicide against the disease. This model presents the idea a revitalization of mass production of *Kapeng barako* in the country, with help of a bio-fungicide. The two objectives of this basic simulation model is to determine if the bacteria can be a biological control agent, where it will suppress the spread of the Coffee Rust disease. And the other is to determined which of the 2 Control Strategies, Global and Local, is the better strategy in inhibiting the Coffee Rust disease growth. The key results of Simulation 9 and 10 have shown the case of *Bacillus subtilis* successfully suppressing the Coffee Rust Disease in spreading out in the plantation and able to control for the disease for long time until it dies due to failed sustenance. It has shown that Global Control is the superior strategy in inhibiting the epidemic disease.

List of Tables

List of Figures

Chapter 1

Introduction

In many years, the beverage Coffee has been vastly produced internationally. The production starts when the seeds from coffee cherries are separated. Then, they will be roasted and grind into fine particles. The simplest way to make the drink is by brewing ground coffee beans in water, preferably hot water(6). Other more sophisticated extraction methods were develop such as Cold brew, French Press, Drip filtration. But they required specialized equipments.

However there are problems and struggles in production such as pests and disease. A common example is the disease named Coffee rust that is caused by the infection fungi *Hemileia vastatrix.* This disease has been a big problem for Coffee Farmers during growing and harvesting the beans. It causes serious defoliation in coffee plants, and as a result, crop yield is diminished. This disease does not immediately kill the plant. In many times in history, this disease has caused big epidemics on coffee plantation around the coffee belt countries where most of the coffee beans are grown(24)

One prevention method in combating the disease is the application of copper based compounds fungicides. They are used to kill the fungi and add a protective barrier to the plant's leaves in order to stop the fungi germination. Copper is also an important micronutrient for the plants' growth and yield(13). However, there is a detrimental effect if used repeatedly. As copper does not easily break down on the soil, this accumulates in the soil which results in toxicity to the plants and other microbial lifeforms. It kills other beneficial bacteria, reduces the soil fertility, and may give more chance for other diseases and pests like Coffee fruit rot and green scale to grow(18). This leads the Coffee rust effect to be weaken but it gives rise more to other coffee plant related diseases to the plant.

There have been alternative methods of using other fungal pathogen such as organic products or fungicides that do not have copper in it. A preferred alternative method is the use of bacteria as biological fungicides. These are bacteria that prevents the growth of diseases due to their chemical composition inhibiting the growth of diseases. One of the known example is the bacteria *Bacillus thuringiensis.* They blocked the germination tubes with their antibiotic compounds, mainly lipopeptides such as iturine, surfactin, and fengicine(9). Thus controlling the rust uredospores. There are also other alternative bacteria which are currently being studied. One of which is the *Bacillus subtillis*, a rod shaped bacteria with a tough protective endospore that is found commonly in soil, humans and marine sponges.

This is a theoretical proposal of an idea of reviving the mass production of the *Barako* with the help of a bacteria fungicide, *Bacilus Subtilis*, to combat the coffee rust. This is done via simulation that is modeled. The model will used also the Hexagonal Planting layout of the coffee plants. The restriction though is that wind is neglected and rain splashes is the only dispersal mechanism. Wind is ignored due to it's random nature and difficult to implement in the model without some restrictions on it.

1.1 Background of the Study

1.1.1 Coffee Liberica and *Kapeng Barako*

Coffea liberica, commonly known as the Liberian coffee, is from the Rubiaceae family. Originally native in Western and Central Africa (Uganda and Angola)(8), it would been naturalized in other countries like Columbia, Venezuela, Indonesia and the Philippines. A unique characteristic of the plant is it can reach up to 18 meters in height. The sizes of it's beans, leathery leaves, and cherries are bigger compared to other coffee varieties. The shape of Liberica bean is also unique. It's asymmetric and much more oblong shape compared to other species. One side is shorter and the other side is longer with a characteristic 'hook' like shape.

Figure 1.1: Shows the physical differences of Arabica, Robusta and Liberica beans. Source from knbk.in.ua

In the Philippines, the provinces of Batangas and Cavite are known to cultivate these coffee plant species. It was by 1740's that Coffee liberica was introduced in the country. There would be a specific variety of Coffee liberica that emerged over the years of growing known as "Kapeng barako" (4). It's consider to be a more intense and bitter version of Liberica with a hint of aniseed flavor. These two provinces led the Philippines to be one of the top producers of coffee in 1880's. But later on that decade, the coffee rust disease destroyed most of the plants, leaving only a handful of barako seedlings. It was then decided that the *Kapeng barako* as an endangered species and should be conserved for cultural reasons(17). Now, it only grows on small farms and is rarely exported. There were multiple times of re-surging interest in mass-production but it didn't proceed. This is due to the big size of the coffee plant and the preference of other coffee-rust resistant cultivars such as *Robusta* and *Catimor hybrids.*

Figure 1.2: Barako beans and coffee. From Hawaii Filipino Chronicle

1.1.2 *Bacillus subtilis*

Bacillus subtilis, also known as the Hay bacillus or Grass bacillus, has a rod shape spore-forming bacteria(12). There are three main physical characteristics that makes this bacteria suitable in biotechnology studies. First, it has a thick layer of peptidoglycan within it's cell wall as a protection layer in extreme temperature environments. Second, it's surface is heavily flagellated that could traverse fluids easily, Lastly, it has a modifiable genetic structure that can be manipulated in laboratory(1).

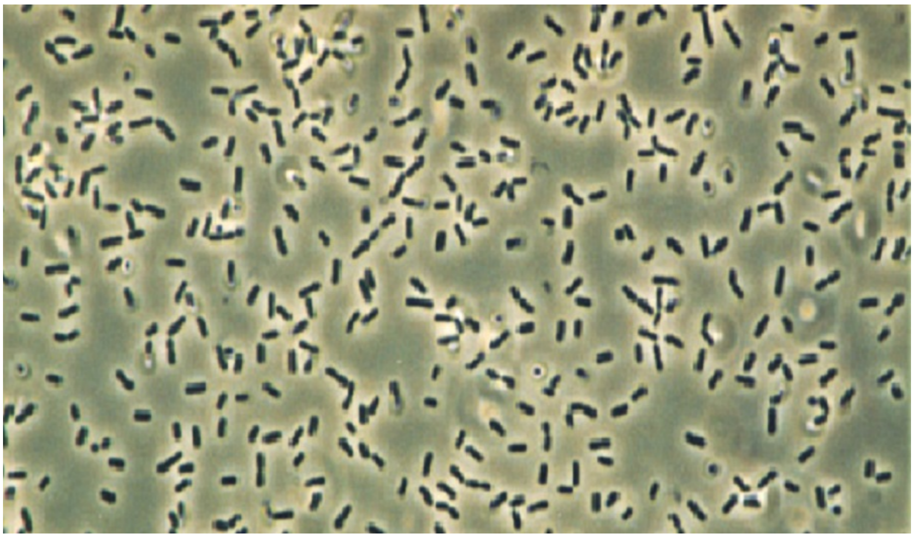


Figure 1.3: A microscopic picture of Bacillus subtilis in laboratory. Source: biology dictionary

Bacillus subtilis can be used as bacteria fungicide and is already in commercially available fungicide products(7). It is used to combat other diseases such as Blight disease, Gray Mold, Damping-Off Disease,etc. When applied to a plant, it forms a biofilm on the plant leaves. This layer of protection inhibits pathogen growth due to bio-composition nature against diseases(27). This makes the bacteria as a bio-controlling agent to plant diseases.

It can hypothetically be used against Coffee Rust disease. In addition of the protection layer earlier, *B. subtilis* also produces also iturins, fengycins, and surfactins(9). These are the same antibiotic compounds produced by another effective bacteria fungicide against Coffee Rust Disease called *bacillus thuringiensis.* These compounds inhibit growth germination pores of the uredospores of *H.vastatrix.* This blocks the germination tubes of the uredospores and the results is a halt of the growth of *H.vastatrix* mycelium. *Bacillus subtilis* has no copper compound in it and as such not toxic to the coffee plants. With all of these, *B. subtilis* can be an alternative bacteria fungicide to *bacillus thuringiensis* and to copper pesticides. It can theoretically be a more effective bacteria fungicide compared *B. thuringiensis* due the former having more protective properties.

1.1.3 Hexagonal Planting

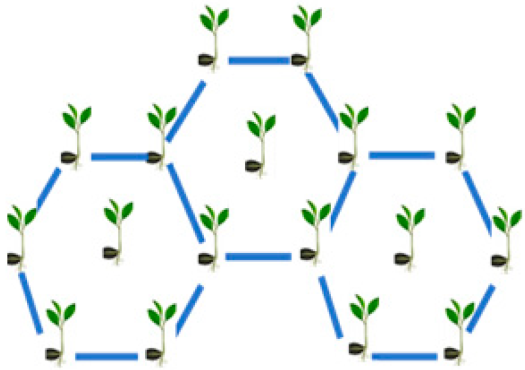


Figure 1.4: An example of Hexagonal Layout. From: Scholarly Community Encyclopedia

Hexagonal Planting is a type of plantation layout where the plants are planted in each corner of an equilateral triangle. Doing this 6 times in a row, creates a hexagonal shape plantation with 6 plants on the corners of the hexagon and a central 7th plant inside of the hexagon(10). This type of planting can be use depending on the plant growing, soil fertilization, size of the plantation, plantation's topography, and the climate of the plantation. There have been examples of crops that uses this plantation layout type like carrots, rice, sunflower, apple trees and also on Coffee plants(22). The advantages of hexagonal planting are: maximizing space, more efficiency in resourced used, more light received by the plants, more quantity in crop yield. There are also disadvantages such as: higher manual labor demand, complex layout planting, possibility of overcrowding, and notably it is more susceptible in spreading diseases quickly.

1.2 Statement of the Problem

Kapeng Barako, once widely cultivated and culturally significant in the Philippines, is now endangered, largely due to the Coffee Rust disease (Hemileia vastatrix). While chemical pesticides such as copper-based fungicides are used to control its spread, they pose sustainability issues and can be toxic at high concentrations.

This study investigated the potential of Bacillus subtilis as a sustainable biocontrol agent for Coffee Rust. It aimed to addressed the following questions:

- Can Bacillus subtilis effectively control Coffee Rust disease?
- What environmental factors influence its effectiveness?

1.3 Objectives of the Study

1.3.1 General Objective of the Study

The primary objective of this study is to simulate the spread of Coffee Rust Disease in a hexagonal plantation of *Kapeng Barako*, using *Bacillus subtilis* as the bacteria fungicide against the disease.

1.3.2 Specific Objectives of the Study

The specific objectives of the study are as follows:

- To evaluate the effectiveness of *Bacillus subtilis* as a bio-fungicide in inhibiting the growth of Coffee Rust Disease in the simulation.
- To investigate how various environmental and simulation parameters affect the bacteria's ability to suppress Coffee Rust.
- To compare the effectiveness of two control strategies—Global Control and Local Control in managing the spread of the disease.

1.4 Significance of the Study

This study proposed a hypothetical model for the mass production of *Kapeng Barako* through the use of a naturally occurring and readily available bio-fungicide, *Bacillus subtilis*. It is intended to serve as a foundational framework for future revitalization efforts aimed at preserving and expanding the cultivation of this culturally significant coffee species. The model serves as a basic theoretical concept of the revitalization plan. It does not fully function realistically since many real life factors such as wind and temperature are not present in it. As such, it is not intended for direct practical application without further empirical validation and research. Direct application of this is highly not advisable.

The proposed revitalization initiative holds potential cultural and economic benefits for the Philippines. This study may also served as a reference for future researchers or stakeholders interested in developing or continuing similar revival projects for local agricultural commodities.

1.5 Scope and Limitation

The scope of this study is geographically limited to the Philippines, specifically the provinces of Batangas and Cavite, as these are the primary regions known for cultivating *Kapeng Barako*. This coffee variety is the central focus of the study, particularly in relation to its vulnerability to Coffee Rust Disease and the potential use of *Bacillus subtilis* as a biological control agent.

This study is conceptual and simulation-based; it serves as a theoretical model rather than a real-world implementation. Therefore, its findings are not intended for immediate practical application without further empirical research.

The model focuses exclusively on Coffee Rust as the antagonistic disease. Other coffee-related diseases such as Coffee Green Scale and Fruit Scale are beyond the scope of this research and are not included in the simulation.

Additional limitations include the exclusion of several real-world environmental factors, such as temperature variability, topography, insect or animal agents, and other ecological influences. In particular, wind effects are omitted due to the complexity of simulating its stochastic nature and challenges in accurately coding wind behavior within the model.

Chapter 2

Review of Related Literature

2.1 Coffee Plant

Coffee (*Coffea arabica*) is a perennial species belonging to the Rubiaceae family, originally native to the southeastern region of the Arabian Peninsula(16). Among various coffee species, *Coffea arabica* is the most widely cultivated and consumed due to its sweeter and milder flavor profile, which appeals to a large global consumer base(3). It is generally more favored than other species such as *Coffea canephora* (commonly known as Robusta) and *Coffea charrieriana*.

However, this desirable flavor comes at a cost *Coffea arabica* is more vulnerable to pests, diseases, and climatic stresses. One of the most significant threats to its production is the Coffee Rust disease, a fungal infection caused by *Hemileia vastatrix*, which can severely reduce yields and damage plantations. *Coffea arabica* is the chosen coffee species in the based Coffee Rust-Bacteria Model(5).

Coffee holds substantial economic and cultural importance in the Philippines. It is both a locally consumed beverage and an exported agricultural product. The introduction of coffee to the country is believed to have occurred around 1696, when the Dutch brought coffee plants to the islands(2).

In Filipino culture, coffee is commonly consumed in the mornings to boost energy and concentration, and in the afternoons to alleviate fatigue or simply as a leisure beverage. In recent years, there has been a noticeable rise in coffee consumption, particularly among students and professionals. This trend is accompanied by the increasing number of coffee shops and local businesses centered around coffee culture, suggesting a continued growth in demand in the foreseeable future.

2.2 Coffee Rust

Coffee Rust is a destructive foliar disease affecting coffee plants, caused by the fungus *Hemileia vastatrix*(23). This pathogen is an obligate parasite that relies exclusively on coffee plants as its host. The disease has been documented in coffee-growing regions across Africa, the Near East, India, Asia, and Australasia. Notably, it was responsible for the collapse of Sri Lanka's once-thriving coffee industry in the late 1800s.

The infection is primarily caused by the germination and spread of *H. vastatrix* uredospores, which infect the leaves of the coffee plant. Initial symptoms typically appear as small, yellowish, oily spots on the upper surface of the leaves. Over time, these lesions expand into larger circular spots that progress in color from bright orange or red to brown, often surrounded by a yellow halo. Severe infestations can lead to premature leaf drop and significant yield loss.

Figure 2.1: Coffee plant leaf infected with an advanced stage of coffee rust. This advanced stage shows an intensified color, which gives yellow or orange spots.

Several environmental and biological factors significantly influence the development and spread of Coffee Rust. These include rainfall, planting density, temperature, and the physiological age of both the uredospores and the host plant's leaves(19). Germination of *H. vastatrix* spores is strongly favored under humid conditions, making moisture a critical component in the life cycle of the fungus.

One of the common countermeasures against Coffee Rust is the application of copper-based fungicides. These compounds inhibit the reproduction of rust spores by interfering with their metabolic processes. However, their long-term use raises environmental and health concerns.

The dispersal of Coffee Rust spores is facilitated by several mechanisms, including rainfall splash, wind currents, gravity, and interactions with animals. High frequency and intensity of these dispersal events can accelerate the spread of the disease across plantations, leading to widespread infection.

2.3 *Bacteria thuringiensis*

Bacillus thuringiensis is a soil-dwelling, gram-positive bacterium widely used as a biological pesticide. It was first discovered by Japanese sericultural engineer Shigetane Ishiwatari and later rediscovered and formally identified by German microbiologist Ernst Berliner.

Today, it is one of the most extensively utilized biological agents in agriculture. It has been employed to manage various coffee plant diseases, including Coffee Green Scale, Coffee Wilt, and Pink Disease. Studies suggest that *B. thuringiensis* can be as effective as copper hydroxide fungicides in controlling Coffee Rust (9). Moreover, it offers a safer alternative with fewer toxicological concerns.

The efficacy of *B. thuringiensis* is attributed to its production of antibiotic compounds—iturin, surfactin, and fengycin—which interact with the germination pores of *H. vastatrix* uredospores. These compounds inhibit the formation of germ tubes, thereby halting the development of mycelium and effectively suppressing fungal growth.

2.4 Model Based

2.4.1 Agent-Based Model

Agent-Based Model is a modeling approach where a number of agents (example: cows, mosquitoes, etc), are place in a system. The goal of this modeling technique is to stimulate and study the actions and interactions between the agents in order to understand the nature of the whole system and determine the outcomes(28). It is stochastic in nature due to the random probabilities of the interaction between agents.The agents are also autonomous and have their own attributes that will contribute in their interactions.

It has been used since 1970's and been continuous used to this day. It's being used in many different kind of fields like Biology, Economics, Social Science, Game Development, Epidemiology,etc. It has been widely used by researchers in focusing a real life systems like ecosystem, business system, etc. It is done by simulating it in digital form, and study it's agents, agents' relationship and other elements.

2.4.2 Population Model

Population model is a type of mathematical-biological modeling that is applied on population dynamics. It can be used to study whether birth rates, mortality rates, length of time, and the interactions between the types of populations affect the populations in the system(11). Several derived models like Lotka–Volterra (Predator and Prey) equations and Nicholson–Bailey (Host-Parasitoid) model are based on population model.

Population Model can be computer simulated for better visualization on population dynamics. It requires mathematical equations in building this model. The more complex the model is, the more complex the equations as well. Simulated population models serve as powerful tools for addressing real-world challenges such as population decline, overpopulation, and resource scarcity. By replicating the dynamics of human or ecological populations, these models enable researchers to generate forecasts of future trends and construct retrodictions to better understand historical events. Such simulations support data-driven decision-making and strategic planning in fields ranging from urban development to environmental management.(15).

2.4.3 Two Population Model

The two-population model is a form of mathematical and biological modeling used to study population dynamics involving the interaction of two distinct groups. These models helps to analyze how factors such as birth rates, death rates, time duration, and inter-population relationships influence the overall system(20).

Well-known derivatives based on the two-population concept include the Lotka–Volterra model, which describes predator-prey dynamics(25), and the Nicholson–Bailey model, which focuses on host-parasitoid interactions(14). These models provide foundational insights into ecological and epidemiological processes, particularly in understanding stability, oscillations, and collapse conditions in population systems.

2.4.4 Coffee Rust-Bacteria Model

Main Equations

This two-population agent-based model investigates the potential of biological control using the bacterium *Bacillus thuringiensis* against Coffee Rust infection, caused by the fungus *Hemileia vastatrix*(5). The simulation is implemented on a rectangular grid representing a plantation of *Coffea arabica* plants. The two primary agent types in the model are the Coffee Rust spores and the bacterial agents.

The model consists of a total of 14 equations. Five of these are core equations that define the main dynamics of the system, including bacterial growth, rust spread, and agent interactions. The remaining equations serve as supporting components—either providing theoretical foundations or representing simplified versions of the main equations for auxiliary processes such as dispersal or control logic.

This section will focus on discussing the five main equations that constitute the central framework of the simulation model.

$$\frac{dB_{i,j}}{dt} = b\,B_{\{i,j\}}\left(1 - \frac{B_{i,j}}{K_B}\right) + \mu_{\{i,j\}}(t) - d\,B_{\{i,j\}},$$
$$\frac{dH_{i,j}}{dt} = b\,H_{\{i,j\}}\left(1 - \frac{H_{i,j}}{K_H}\right) + \alpha\,I_{\{i,j\}} - \left(B_{\{i,j\}} + \beta\right)H_{\{i,j\}}$$

Equation 1 is a system of differential equations that describes the growth and interactions of both the bacteria (B) and coffee rust (H) populations on each plant across the plantation. These equations account for the logistic growth of each population, $dB_{i,j}$ and $dH_{i,j}$, as well as the effects of the bacteria on the coffee rust and the dispersal of both populations.μ represents the irrigation rate of new bacteria being placed on a plant, depending on the strategy used. K_B and K_H are the carrying capacities of the bacteria and rust spores, respectively. α is the immigration rate of new rust spores, and β is the emigration rate of rust spores. It is the equation of the number of rust spores that can go from the neighboring plant to the chosen which will be discussed next.

$$I_{\{i,j\}} = \left(\sum_{k=-1}^{1}\sum_{l=-1}^{b} H_{\{i+k,j+l\}}(t)\,\tau_{\{i+k,j+l\}}\right) - H_{\{i,j\}}\,\tau(0,0,t)$$

Equation 2 is defined as a random variable dependent on the coffee plant's neighbors, which correspond to the plants in the cardinal and intermediate directions of the (i,j)th plant. This function represents the number of spores that can travel from any neighboring coffee plant to the (i,j)th plant. It's worth mentioning that process involves summation of rust spores from neighboring plants,then minus the sum with the rust spores of the (i,j)th. It is incorrect to interpret that each difference of the number of rust spore between a neighboring plant and (i,j)th. Then, summation those differences. The results will be to large and illogical that even normalization would not fix this.This makes it unrealistic and unreliable. $\tau(k,l,t)$ represents the random variable for the proportion of coffee rust spores $H_{i,j}$ that are transferred from plant (i,j) to plant $(i+k, j+l)$. This variable follows a Bernoulli distribution, as described by Equation 3.$\tau(k,l,t) \sim Bernoulli(P_{i,j}(k,l,t))$

$$P_{\{i,j\}}(k,l,t) = \psi\lambda(t)H_{i,j}(t)D(||(k,l)||_{\{i,j\}})$$

$$\mu_{\{i,j\}}(t) \equiv \mu,$$

$$\mu_{\{i,j\}}(t) = \rho\, H_{\{i,j\}}(t) + \delta\left(\sum_{k=-1}^{1}\sum_{l=-1}^{1} H_{\{i+k,j+l\}}(t) - H_{\{i,j\}}(t)\right).$$

Equations 4 and 5 define the bacterial irrigation rates based on the control strategy implemented in the simulation. Two control strategies are considered: Global Control and Local Control.

In the Global Control Strategy, bacterial irrigation is applied uniformly across the entire plantation. This proactive approach aims to prevent the early spread of infection by ensuring that each plant receives the same intensity of bacterial application. Mathematically, the irrigation rate is constant for all plants regardless of their infection status.

In contrast, the Local Control Strategy applies irrigation selectively based on infection levels. This reactive approach increases bacterial concentration on plants that are more severely infected, while neighboring plants receive a reduced dosage. In this context, ρ denotes the irrigation intensity for the most infected plant(s), and δ denotes the irrigation intensity for its adjacent neighbors with lower infection levels. If multiple plants share the highest level of infection, they are collectively treated as the primary targets and each receives the ρ rate of bacterial application.

This differentiated irrigation strategy reflects a targeted response that adapts to the infection dynamics within the plantation.

The parameters of the model are given:

Parameter	**Description**	**Value(s)**	**Actual values**
b	Natural growth rate of bacteria	$2\,h^{-1}$	2
K_B	Carrying capacity of bacteria	$7\log$ bacteria	7
d	Natural death rate of bacteria	$1.5\,h^{-1}$	1.5
K_H	Carrying capacity of coffee rust	{0.2380, 0.2840} rust proportion	0.2380 or 0.2840
h	Natural growth rate of rust	$(0, 0.35)\,h^{-1}$	0–0.35
α	Immigration rate of coffee rust	$(0, 1)\,\mathrm{day}^{-1}$	0 or 1
β	Emigration rate of coffee rust	$(0, 1)\,\mathrm{day}^{-1}$	0 or 1
ψ	Proportionality constant of dispersal probability	$[0, 20]\,(\text{rust proportion})^{-1}\,\mathrm{m}$	0 - 20
γ	Conversion rate of bacteria to coffee rust	$(0, 1)\ \log(\text{bacteria})^{-1}\,\mathrm{day}^{-1}$	0 or 1
σ	Standard Deviation of dispersal probability of the model	$3\,\mathrm{m}^{-1}$	3
$d_{ij}(0, \pm 1)$	Distance between columns of coffee plants	$2\,\mathrm{m}$	2
$d_{ij}(\pm 1, 0)$	Distance between rows of coffee plants	$1\,\mathrm{m}$	1
$d_{ij}(\pm 1, \pm 1)$	Diagonal distance between coffee plants	$2.2361\,\mathrm{m}$	2.2361

Table 2.1: Parameter values and descriptions of the model.

The adjustable parameters within of the model are: K_B, K_H, α, β, ψ, and γ. There are also adjustable that can be chosen by the user like the number of rows,number of columns, infected proportion, type of control strategy along with there respective irrigation rates which are: μ in main Global Control Strategy and ρ and δ for the Local Control Strategy.

Numerical Simulations

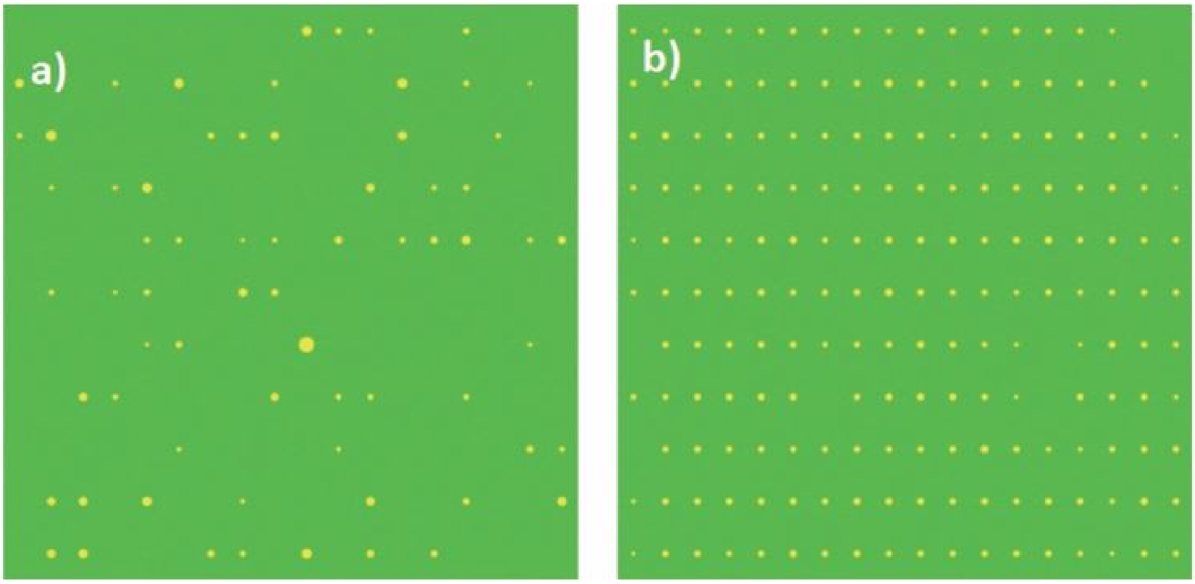


Figure 2.2: Example Simulation of the model in Netlogo. A top view of an example plantation. Showcasing the (a)initial state of the plantation and the (b)end state of the plantation at specific end of the simulation hour mark

The model equations were implemented and programmed using the NetLogo simulation environment. A graphical representation of the simulation is shown above from a top-down perspective. In this visualization, green patches represent healthy coffee plants, which are evenly distributed across the plantation. Yellow dots indicate infected plants, with the size of each dot corresponding to the proportion of Coffee Rust infection—the larger the dot, the higher the level of infection present in that plant.

The figure consists of two simulation snapshots: Figure 2.2.a depicts the initial state of the plantation at the start of the simulation, while Figure 2.2.b shows the final state after 48 simulation ticks, equivalent to 48 hours of simulated time. This duration was selected as the standard runtime for the initial experiment. Once the simulation reaches this time threshold, it automatically pauses.

Although 48 hours serves as the default runtime, the simulation can be extended beyond this limit at the discretion of the researcher. This is called Further Simulation Run. In such cases, the model will continue running indefinitely until manually stopped by the user.

2.4.5 Summary

In summary, this study developed an agent-based model to simulate the infection control capabilities of the bacterium *Bacillus thuringiensis* against the Coffee Rust disease, which is caused by the fungus *Hemileia vastatrix*. The model incorporates equations that approximate a rain splash-based dispersal mechanism as realistically as possible. The coffee plants in the simulation are arranged in a uniformly distributed rectangular grid, representing the plantation layout.

Two control strategies were implemented in the model: Global Control and Local Control. Only one strategy may be selected per simulation run. In the Global Control Strategy, all plants receive an equal level of bacterial irrigation. In contrast, the Local Control Strategy applies a higher irrigation rate to infected plants while their neighboring plants receive a reduced dosage.

Each simulation runs over a 48-hour period within the simulation. After completion, the resulting data primarily visualized through generated graphs is analyzed to determine which parameters influence the interactions between the bacterial agents and rust spores. These findings help evaluate the effectiveness of each strategy under varying conditions.

Chapter 3

Methodology

This section discusses the mathematical nature of this study's model. This where the equations, parameters and explanations of the components of the model.

3.1 Model

The model is described as a system of non-linear differential of equations. It is an agent-based model where the agents are: Coffee Plant/plants, Coffee Rust Spores and the bacteria *Bacillus subtilis*. The model is mostly based on the previous model mentioned in the Review Related Literature(5) and is closely related Nicholson–Bailey model(host-parasitiod).

The coffee plantation is a hexagonal plantation layout. Coffee Rust spore and Bacteria populations are still both denoted by $H_{\{i,j\}}$ and $B_{\{i,j\}}$ respectively. Both of these populations grow logistically, where b and h represent the growth rates. K_B and K_H are the carrying capacities of bacteria and rust spores. The model is then described by the system of non-linear differential of equations:

$$\frac{dB_{\{i,j\}}}{dt} = b\, B_{\{i,j\}} \left(1 - \frac{B_{\{i,j\}}}{K_B}\right) + \mu_{i,j}(t) - d\, B_{\{i,j\}},$$
$$\frac{dH_{\{i,j\}}}{dt} = h\, H_{\{i,j\}} \left(1 - \frac{H_{\{i,j\}}}{K_H}\right) + \alpha\, Q_{\{i,j\}} - \big(B_{\{i,j\}} + \beta\big)\, H_{i,j}.$$

where:

- $H_{\{i,j\}}$ - rust spore population on a plant
- $B_{\{i,j\}}$ - bacteria population on a plant
- b - bacteria growth rate
- d - bacteria death rate
- h - rust spore growth rate
- $\mu_{\{i,j\}}(t)$ - irrigation rate of new bacteria
- α - Immigration rate of coffee rust spores
- β - Emigration rate of coffee rust spores
- $Q_{\{i,j\}}$ - the Q function. It denotes the possible number of spores that can travel to the [i,j] from it's neighboring plants.

We need a function that defines the number of spores that can go from any neighboring coffee plants to {i,j}th plant. We named this function as Q Function. This is an approximates the number of spores that can go to the {i,j}th plant, not the exact number of spores. This is done by sum-mating the total rust proportion of the 6-neighboring plants around the {i,j}th plant first. Then, minus the total with the {i,j}th plant's rust proportion. This essentially replicates a windless rust spore spread approach. It should not interpret as: Subtract the rust proportion of each of the 6-neighboring plant to the {i,j}th each. Then totaling those differences. It is incorrect as this ignores the spatial factor of the dispersal mechanism and the outcome is mostly illogical and too big within the model, even if normalization is applied.

$$Q_{\{i,j\}} = \left(\sum_{k=-1}^{1} \sum_{l=-1}^{b} H_{i+k,j+l}(t)\, \tau_{i+k,j+l} \right) - H_{\{i,j\}}\, \tau_{i+k,j+l}(0,0,t)$$

where:

- $H_{\{i,j\}}$ - rust spore population on a plant
- $\tau_{\{i+k,j+l\}}$ - random variable that represents the proportion of coffee rust spore $H_{\{i,j\}}$ that is passed from plant {i,j} to plant {i+k,j+l}.
  - $\tau_{\{i+k,j+l\}}(0,0,t)$ - rust proportion on the i,j plant at time t
  - $\tau_{\{i+k,j+l\}}(k,l,t)$ - rust proportion on the {i+k,j+l} or any of the 6-neighboring plants.k and l represent the direction/cartesian coordinates.
- $\left(\sum_{k=-1}^{1} \sum_{l=-1}^{b} H_{i+k,j+l}(t)\, \tau_{i+k,j+l} \right)$ - the rust proportion summation with Rust populations of the 6-neighboring plants around the i,j plant
- $H_{\{i,j\}}, \tau_{\{i+k,j+l\}}(0,0,t)$ - rust proportion with Rust population on the {i,j} plant

The value of $\tau_{(i+k,j+l)}$ is a Bernoulli variable value. It means that it's value must be either 1 or 0. With 1, indicating that the rust spore guarantees to disperse to that plant and 0 means that the rust spore will not disperse on that plant. It is independent variable but is probabilistic affected by certain other variables such as and distances between plants. The following forms of $\tau_{(i+k,j+l)}$ correspond to certain directions

- $\tau_{(-1,1)}$ - upper-left diagonal direction
- $\tau_{(1,1)}$ - upper-right diagonal diagonal direction
- $\tau_{(-1,0)}$ - left direction
- $\tau_{(1,0)}$ - right direction
- $\tau_{(-1,-1)}$ - lower-left diagonal direction
- $\tau_{(1,1)}$ - lower-right diagonal direction

The Control Strategies serve as counteraction plan against the Coffee Rust Disease. This is done by the application of the bacteria *Bacillus subtilis* on Coffee Plants. There are 2 types of Control Strategies. Only one Control Strategy can be applied at a time; simultaneous use of two strategies is not allowed. They are both represent by the parameter $\mu_{\{i,j\}}$ but differ in the formulation of $\mu_{i,j}$ depending on the selected control strategy. The first type is the Global Control. This control strategy defines that the irrigation of new bacteria is the same amount intensity throughout the plantation. This means that the intensity of spraying the bacteria fungicide is homogeneous on all plants. It is a passive strategy where it's goal is to prevent the rust spore spread ahead of time. This means for all i, j,the formulation of $\mu_{i,j}$ is:

$$\mu_{i,j}(t) \equiv \mu,$$

Now for the second type is called Local Control Strategy, it is define that the irrigation of new bacteria differs between plants in terms of intensity. There is so called 'target plant' in the plantation. This is the plant with the highest infection level in the plantation. The intensity spray on that target plant is greater than the intensity spray on it's neighboring plants. It is a heterogeneous and reactive in nature compared to Global Control Strategy. ρ represent the intensity spray on the target plant and δ represent the intensity spray on the target plant's neighbors. If there is a case where there are multiple adjacent plants with the same highest infection level, they are treated as one entity and therefore receive the same intensity spray. The target plant is updated every hour if the infection level of the previously targeted plant has decreased..

$$\mu_{\{i,j\}}(t) = \rho\, H_{\{i,j\}}(t) + \delta \left(\sum_{k=-1}^{1} \sum_{l=-1}^{1} H_{\{i+k,j+l\}}(t) - H_{\{i,j\}}(t) \right).$$

where:

- ρ - irrigation rate on the infected [i,j] plant
- δ - irrigation rate on the 8-neighboring plants around infected [i,j] plant
- $H_{\{i,j\}}(t)$ - rust spore population of infected [i,j] plant at time t
- $\left(\sum_{k=-1}^{1} \sum_{l=-1}^{1} H_{\{i+k,j+l\}}(t) - H_{\{i,j\}}(t) \right)$ - rust spore population of the 8-neighboring plants around [i,j] plant

3.2 Algorithm Design

This section discussed the algorithm, function progress, and complexities of the model. There are algorithms and flowcharts that explained each functions of the simulation model(21).

General Flow/Pseudo-Code of the Model

This is the pseudo-code of the General Flow of the model. It describes the process of using the model in both outside and inside to the user. Below of it is an equivalent graphical flowchart. This is for the ease of understanding more the pseudo-code in a visual way.

Algorithm 1: General Flow of Infection Model for Coffee Rust Using Bacteria

```
Initialize Environment
for each tree (i, j) in plantation grid do
    Initialize rust population H[i][j]
```

    Initialize bacteria population $B[i][j]$
    Set initial infection and bacterial application rates
end for
Define Parameters
Set constants: $b, d, h, \gamma, \alpha, \beta, \psi, K_B, K_H, \sigma$
Define rainfall probability $P_\lambda(t)$
Define dispersal function $D(x)$ (e.g., Gaussian)
for each timestep $t = 0$ to T **do**
    for each tree (i, j) **do**
        Update bacteria population:
        $B[i][j] \leftarrow B[i][j] + b \cdot B[i][j] \left(1 - \frac{B[i][j]}{K_B}\right) + \mu[i][j] - d \cdot B[i][j]$
        Calculate dispersal $I[i][j]$ from neighbors:
        for each neighbor (k, l) **do**
            if Rainfall occurs (Bernoulli($P_\lambda(t)$)) **then**
                $I[i][j] \leftarrow I[i][j] + \tau \cdot H[k][l]$
            end if
        end for
        Update coffee rust population:
        $H[i][j] \leftarrow H[i][j] + h \cdot H[i][j] \left(1 - \frac{H[i][j]}{K_H} + I[i][j]\right) - \gamma \cdot B[i][j] \cdot H[i][j]$
        Update control strategy:
        if Global strategy **then**
            $\mu[i][j] \leftarrow \mu$
        else
            $\mu[i][j] \leftarrow \rho \cdot H[i][j] + \delta \cdot$ sum of neighbor rust levels
        end if
    end for
end for
Analyze Results
Calculate equilibrium and basic reproductive number R_0
Determine system stability (disease-free or endemic)

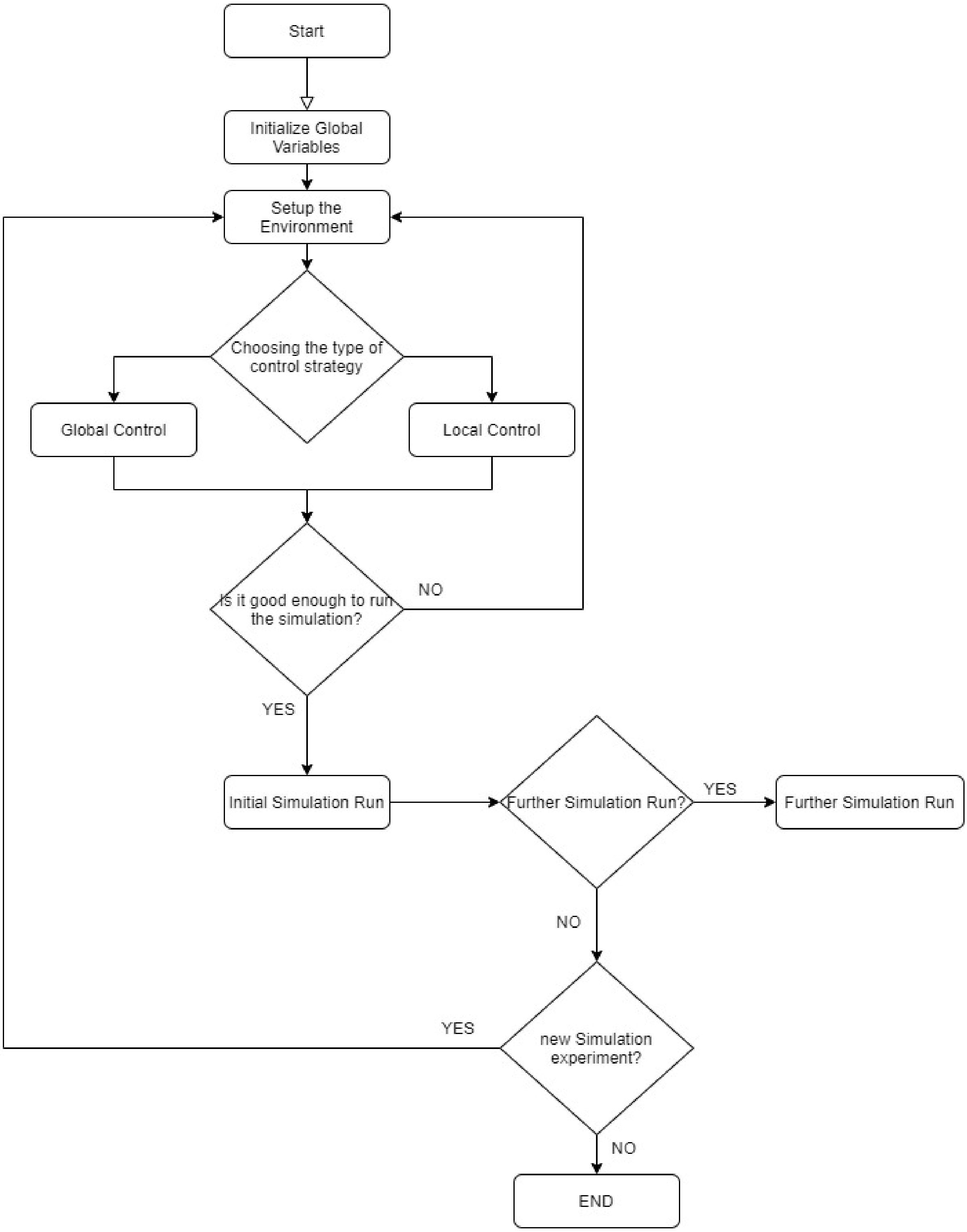


Figure 3.1: General Flow/Pseudo-Code of the Model

Setting up the Plantation

This function creates the patches of the plantation. It setups patches to green. Green indicates a healthy/non-infected plant.

Algorithm 2: Setup Turtles

procedure SETUP-PATCHES
    Set patches color to green
end procedure

Setting up the turtles

This function makes the turtles. It first clears any existing turtles and patches. This resets the plantation. It then determines the grid size of the plantation from the given number of rows and columns. It then creates the turtles.A loop is done where it assigns the turtles to their given patches in the grid. A hexagonal planting layout will be done where there are 7 turtles(Coffee plants). 6 represents the sides of the hexagonal while the 1 will be at the center of the hexagonal. If the turtle's row is an even number, then it offsets that turtle to the right. This action creates the hexagonal layout. It then sets the initial coffee rust level given from the user as 'infected proportion level' and will randomize choose which turtles will be the 'infected'. It will then sets up those 'infected' turtles color to yellow. Yellow signifies as infected plant status. The bigger the size of the yellow circle, it implies the higher the infection level of that plant.

Algorithm 3: setup-turtles Procedure

procedure SETUP-TURTLES
    Clear all existing turtles and reset environment
    Determine plantation grid size $n \times m$
    Create $n \cdot m$ turtles ▷ One per tree
    for all turtles **do**
        Assign turtle to a unique patch in the grid
        Set initial coffee rust level $H[i][j]$
        if random chance ¡ initial infection rate **then**
            $H[i][j] \leftarrow$ non-zero value
        else
            $H[i][j] \leftarrow 0$
        end if

Set initial bacteria level $B[i][j] \leftarrow 0$
Set visual appearance based on infection status
end for
end procedure

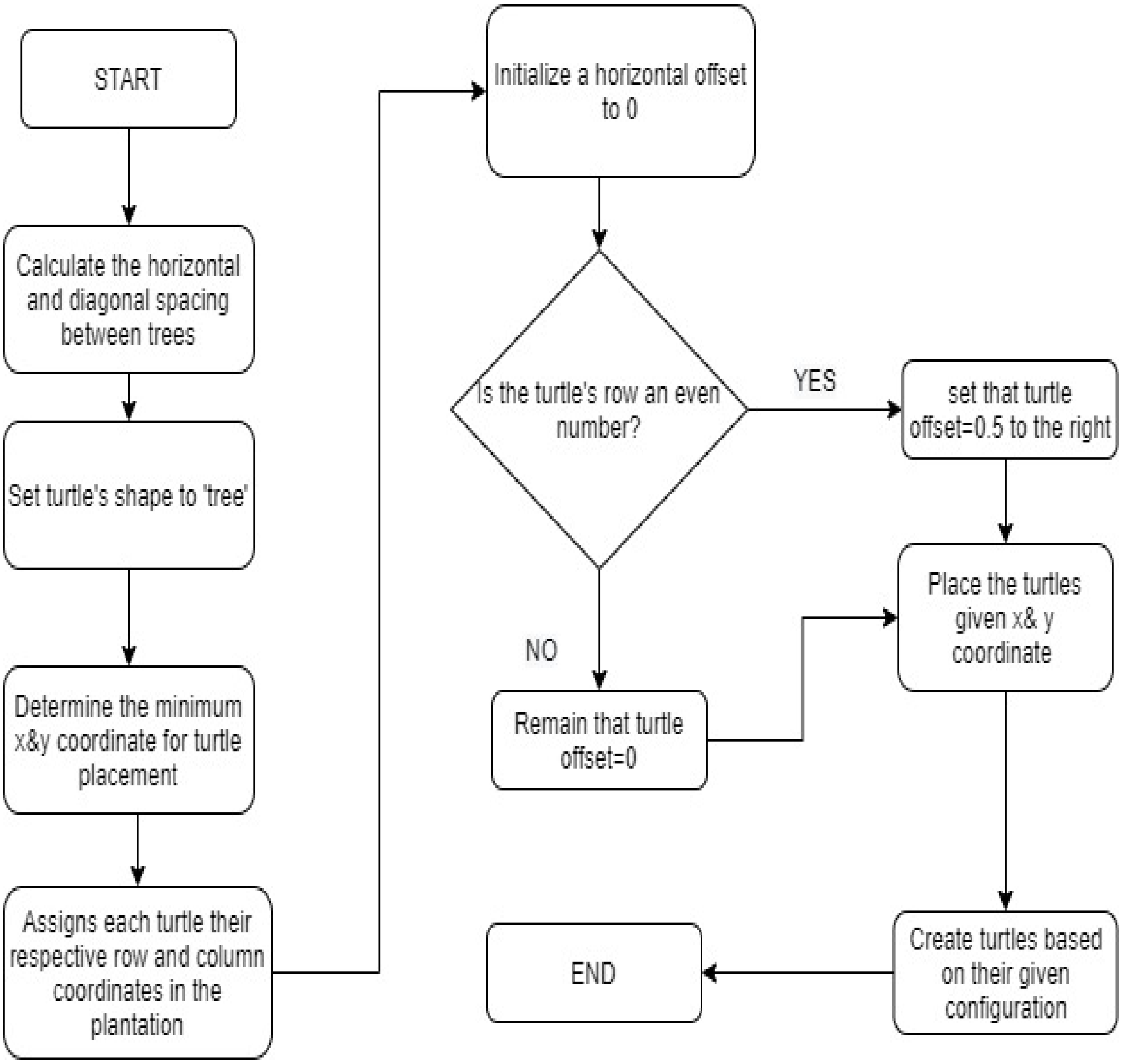


Figure 3.2: Hexagonal Plantation Process Modeling

Report Proportion

This is the Report Rust Proportion function on all of the turtles. It tells the current rust proportion on a plant. It starts by initializing a variable to store the sum of proportions of all turtles it then returns the total proportion value of all turtles. This function repeats for every hour.

Algorithm 4: report-proportion Procedure

procedure REPORT-PROPORTION
  Initialize S variable- the variable to store the sum of proportions
  Sums the proportion values of all turtles
  Returns the total proportion value
end procedure

Dispersal Event Process

This function recreates the rain splash dispersal mechanism. This is the only dispersal mechanism for the Coffee Rust Spore to traverse plant to plant. It starts by initializing the 6 direction variables to 0. Then a loop function on the turtles is used. It computes the directional probabilities using a Gaussian Function, then have a randomization of which direction that the rust spore will follow. This loop function direction starts from the top-left, goes to the right direction until it reaches the bottom right. Then finally an update to the rust proportion of the plant will be done next since the plants receives new rust spores.

Algorithm 5: disperse-rust Procedure

procedure DISPERSE-RUST
  for all patches (i, j) in the grid **do**
    if patch (i, j) has rust **and** rain occurs (Bernoulli(P_λ)) **then**
      for all neighboring patches (k, l) within splash radius **do**
        Calculate distance $d \leftarrow$ distance($(i, j), (k, l)$)
        Compute dispersal probability $p \leftarrow \psi \cdot D(d)$ ▷ Gaussian function
        Draw random number $r \sim$ Uniform$(0, 1)$
        if $r < p$ **then**
          Determine transfer amount: $\Delta H \leftarrow$ proportion of rust from (i, j)
          $H[i][j] \leftarrow H[i][j] - \Delta H$
          $H[k][l] \leftarrow H[k][l] + \Delta H$

end if
end for
end if
end for
end procedure

Control Strategies

This is the Control Strategy Function of the model. It asks first a binary question: 'Is Global Control Strategy selected?'. The next action depends on the answer to that binary question.

- If YES, Global Control Strategy will be chosen and the bacteria-size of irrigation are the same amount based from the user's input of 'main-irrigation'. μ is the used irrigation rate here.
- If NO, Local Control Strategy will be chosen and adjust the bacteria-size of irrigation of the turtle based on the formulation of μ in Equation 5. ρ and δ are the used irrigation rates here.

With Global Control Strategy, it ends at that step. On Local Control Strategy, there are additional steps. A loop function on all plants is used where it determines the most infectious plants in the plantation. Then it recomputes the irrigation rate and applies the given output irrigation. This loops continues and will not stop until the initial simulation ends or forcibly stop by the user.

Algorithm 6: Control-strategy Procedure

procedure CONTROL-STRATEGY
  if Global control is active **then**
    for all patches (i, j) in the grid **do**
      Apply constant bacteria level: $\mu[i][j] \leftarrow \mu$
    end for
  else if Local control is active **then**
    for all patches (i, j) in the grid **do**
      $R_{\text{self}} \leftarrow H[i][j]$ ▷ Rust on current tree
      $R_{\text{neighbors}} \leftarrow \sum H[k][l]$ for all neighbors (k, l)
      Compute local bacteria application rate:
      $\mu[i][j] \leftarrow \rho \cdot R_{\text{self}} + \delta \cdot R_{\text{neighbors}}$
      Apply bacteria based on $\mu[i][j]$
    end for
  end if

end procedure

Growing Rust Spores Function

The rust spore grows with the said formula in 5.5.1. The rust spore grows first then the rust dispersal function proceeds. After the proportion of rust is set. A temporary proportion value is set to 0 to avoid the getting negative values. This function is done per simulation hour.

Algorithm 7: Rust Growth Procedure

procedure GROW-RUST
  Ask turtles to:
  Update `proportion` using:

$$\texttt{proportion} \leftarrow \texttt{proportion}$$
$$+\texttt{proportion-temp} \cdot \texttt{growth-rate-rust} \cdot \left(1 - \frac{\texttt{proportion-temp}}{\texttt{carrying-cap-rust}}\right)$$
$$-\texttt{emmigration-rate} \cdot \texttt{proportion-temp}$$
$$-\texttt{conversion-rate} \cdot \texttt{bacteria-size} \cdot \texttt{proportion-temp}$$

▷ Logistic growth, emigration, and bacteria conversion

  Call `disperse-rust` ▷ Spreads rust across turtles or patches
  Ask turtles to:
  Set `proportion` to `max(0, proportion)` ▷ Prevent negative values
  Set `proportion-temp` to `proportion` ▷ Update for next step
end procedure

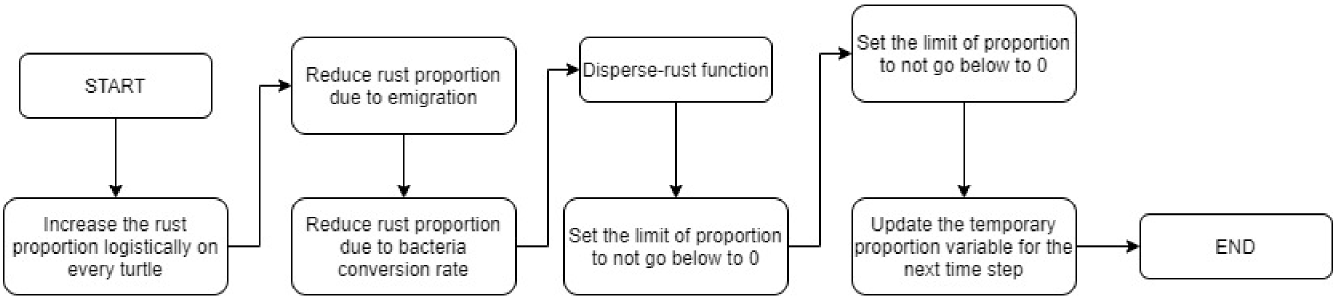


Figure 3.3: Growth of Coffee Rust Spore over time process

Growing Bacteria Function

This is the bacteria growth function based on the equation in 5.5.1. Bacteria grows from the given formula. Then addition of new bacteria from irrigation is then added next. This function is done per simulation hour as well.

Algorithm 8: Bacteria Growth Procedure

procedure GROW-BACTERIA

Ask turtles to:

Update `bacteria-size` using logistic growth model:

$$\texttt{bacteria-size} \leftarrow \texttt{bacteria-size} + \texttt{bacteria-size} \cdot \left(\texttt{growth-rate-bacteria} \cdot \left(1 - \frac{\texttt{bacteria-size}}{\texttt{carrying-cap-bacteria}}\right) - \texttt{death-rate}\right)$$

Ask turtles to apply `control-strategy`

end procedure

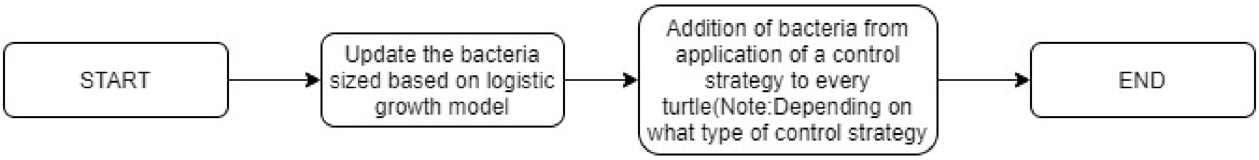


Figure 3.4: Growth of Bacteria over time process

Go Simulation Function

This is the Go function of the simulation. Once the 'Go' button is pressed,the Initial Simulation will be run.it updates the size of each turtle first, based on its proportion value. Then it will call grow-rust and grow-bacteria functions. These two functions are responsible in simulating the growths of the bacteria and rust spores. The Initial Simulation Run will stop when the tick counter reaches 48, which corresponds as 48 hour mark in real life. While the Further Simulation starts after Initial Simulation and will not stop simulating until forcibly stopped by the user.

Algorithm 9: Go Simulation Function

procedure GO SIMULATION FUNCTION
    Ask turtles to set size to `proportion * 3`
    Call `grow-rust` ▷ Simulate rust growth
    Call `grow-bacteria` ▷ Simulate bacteria growth
    Advance time step with `tick`
    if ticks = 48 **then**
        Stop simulation
    end if
end procedure

3.3 Simulation

Several simulations experiments can be conducted after the implementation of the equations and the function algorithms. Netlogo is the chosen simulation program in this study. In order for the model to be used for studying the agents's dynamics, adjustable parameters must be made and have value range. These are the adjustable parameters where their values can be within the code, not visible to the user:

- b - growth rate of the bacteria
- K_H - Carrying capacity of coffee rust spore
- α - Immigration rate
- β - Emigration Rate
- γ - Conversion rate of bacteria
- ψ - Proportionality constant of dispersal probabilities

There are also parameters that are adjustable within the user interface and are given by the user:

- Number of Columns
- Number of Rows
- Infected Proportion on the plantation
- Main Irrigation rate - irrigation rate in Global Control Strategy. Also serves as ρ, the initial irrigation rate on the infected plant in Local Control Strategy
- Secondary irrigation rate - the δ in Local Control Strategy, the irrigation rate on the infected plant's neighbors in Local Control Strategy. Only usable in Local Control Strategy.

There are some differences in this model parameter's value and those from Coffee Rust-Bacteria Model. The vertical direction and their distances are remove due the layout being hexagonal. It only considers the row(horizontal) and diagonal distance and their dispersal direction. Other minor differences lies in the range of some parameters given below.

Parameter	**Value / Range**	**Description**
Maximum number of rows and columns	40×40	Increased grid size to accommodate the new plantation layout.
ψ (Proportionality constant of dispersal probability)	[0–50]	Adjusted due to changes in plant spacing and to reflect increased efficiency of disease spread.
K_H (Carrying capacity of coffee rust)	0.2350 or 0.300	Modified upper bound for improved consistency with observed rust dynamics.
h (Natural growth rate of coffee rust)	[0–0.75]	Increased to reflect greater disease susceptibility under the new plantation structure.
d (Natural death rate of bacteria)	[1.0–1.5]	Changed due to *Bacillus subtilis* being more resistant and having greater survival than *Bacillus thuringiensis*.
$d_{ij}(\pm 1, 0)$ (Horizontal distance between coffee plants)	4 m	Based on the recommended row spacing for optimal planting density.
$d_{ij}(\pm 1, \pm 1)$ (Diagonal distance between coffee plants)	5 m	Based on recommended diagonal spacing between coffee plants.

Table 3.1: Summary of Modified Parameters and Plantation Layout Settings

Below is the new list of parameters and their values for the model.

Parameter	**Description**	**Value(s)**	**Actual values**
b	Natural growth rate of bacteria	$2\,h^{-1}$	2
K_B	Carrying capacity of bacteria	$7\log$ bacteria	7
d	Natural death rate of bacteria	$1.0\,h^{-1}$	1.0
K_H	Carrying capacity of coffee rust	{0.2350, 0.300} rust proportion	0.2350 or 0.300
h	Natural growth rate of rust	$(0, 0.75)\,h^{-1}$	0-0.75
α	Immigration rate of coffee rust	$(0, 1)\,\mathrm{day}^{-1}$	0 or 1
β	Emigration rate of coffee rust	$(0, 1)\,\mathrm{day}^{-1}$	0 or 1
ψ	Proportionality constant of dispersal probability	$[0, 50]$ $(\text{rust proportion})^{-1}\,\mathrm{m}$	0–50
γ	Conversion rate of bacteria to coffee rust	$(0, 1)$ $\log(\text{bacteria})^{-1}\,\mathrm{day}^{-1}$	0 or 1
σ	Standard Deviation of dispersal probability of the model	$3\,\mathrm{m}^{-1}$	3
$d_{ij}(\pm 1, 0)$	Distance between rows of coffee plants	$4\,\mathrm{m}$	4
$d_{ij}(\pm 1, \pm 1)$	Diagonal distance between coffee plants	$5\,\mathrm{m}$	5

Table 3.2: New Parameter values and it's descriptions of the new model.

With these equations and algorithms, the model is constructed and can be implemented in programming languages. This can coded to any programming languages that supports agent based model simulation. Different simulation experiments can now be do next by changing the parameters values and their results can be examined.The next section are the results and discussion of a set of simulation experiments conducted.

Chapter 4

Results and Discussion

A total of 10 different kinds of simulation experiments. Each of these 10 simulations have different parameter configuration. The last 2 simulations are special because the 2 types of Control Strategy are both use nonsimultaneously for comparison of the two control strategies. They will be run for 48 hours simulation hours. The chosen programming language to simulate the model in this study is Netlogo(26). The results data of each simulation will be showed with 4 graphs: The initial and 48 hour mark state of the plantation, Proportion of infected plants and Proportion of rust. These results will be discussed and the parameters effect will be examined.

4.1 Simulation 1

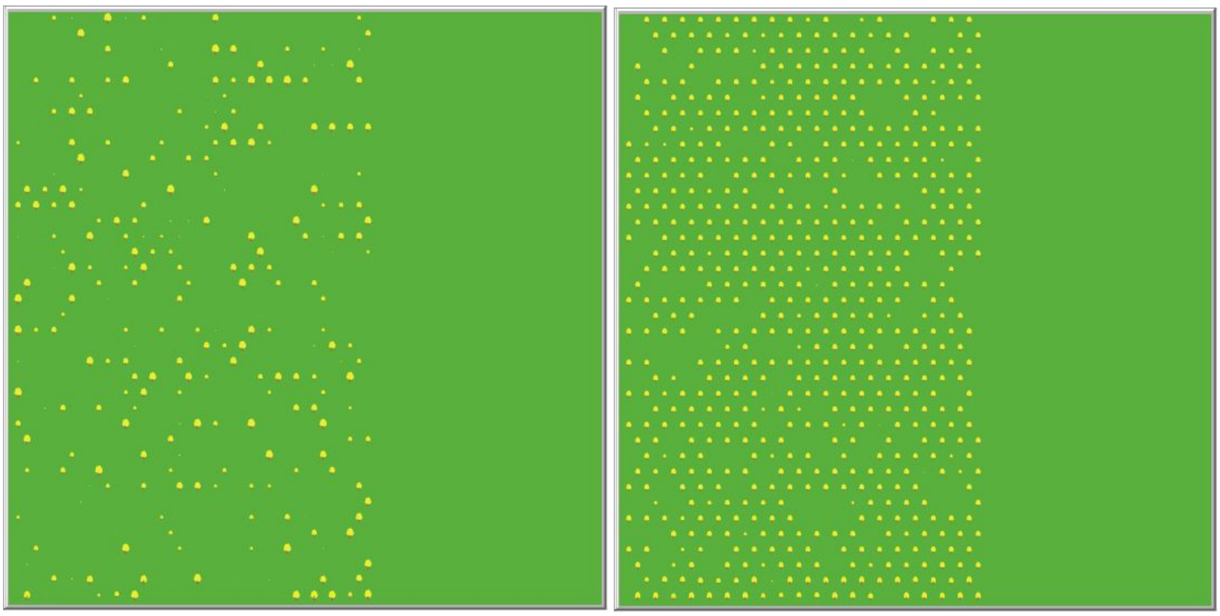


Figure 4.1: Simulation 1: Time series output of 45% Infection Proportion simulation using $h = 0.40$, $K_H = 0.2550$, $\rho = 1$, $\delta = 0.9$,$\alpha = 0.8$, $\beta = 0.6$, $\gamma = 1$, $\psi = 20$ in a 20x40 plantation using the Local Control Strategy.

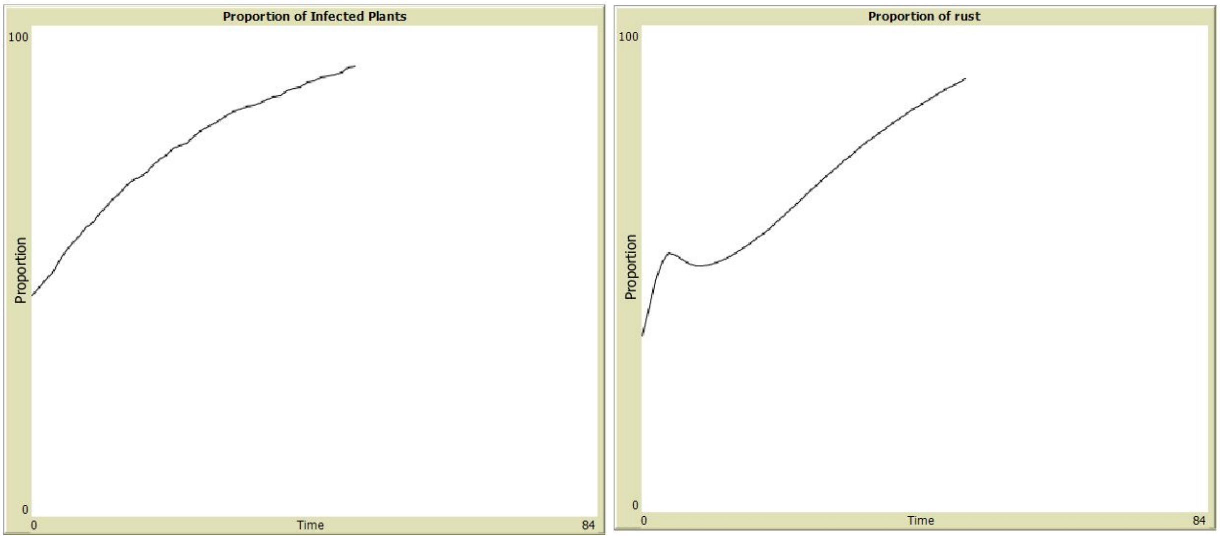


Figure 4.2: Proportions of infected and rust in Local Control Strategy in Simulation 1.

In Simulation 1, we have 45% Infection Proportion, $K_H = 0.2550$, a 1 on main irrigation rate, 0.9 for the second irrigation rate, 0.8 in immigration rate, 0.6 in emigration rate, 1.0 in the bacteria conversion rate and a 20 proportion constant in 20x40 plantation. Local Control Strategy is the chosen strategy here. This simulation has medium Infection Proportion, completely 100% in the irrigation rate, The resulting phenomenon is a the bacteria unable to suppress the coffee rust growth. This is due to high growth rate, immigration and emigration rate that make the rust disease to spread fast across the plantation. The end of initial simulation shows many of the plants to have the same infection level and it's an average level. There is a slight decrease of rust proportion as shown in the Proportion of rust but eventually the rust disease bypass the bacteria's conversion rate, even a 100% conversion rate value. It result to some plants have high infection level at the start of the simulation,to be lessen as time goes on. This is evidently shown on the sizes of the yellow circles in the plantation. The bacteria seems to make all plants to have the same infection level instead. This simulation makes the bacteria not a viable and assuring biological control agent against the Coffee Rust disease. Continuing the initial simulation will result the plantation to be fully infected in just nanoseconds with all plants having the same average infection level.

4.2 Simulation 2

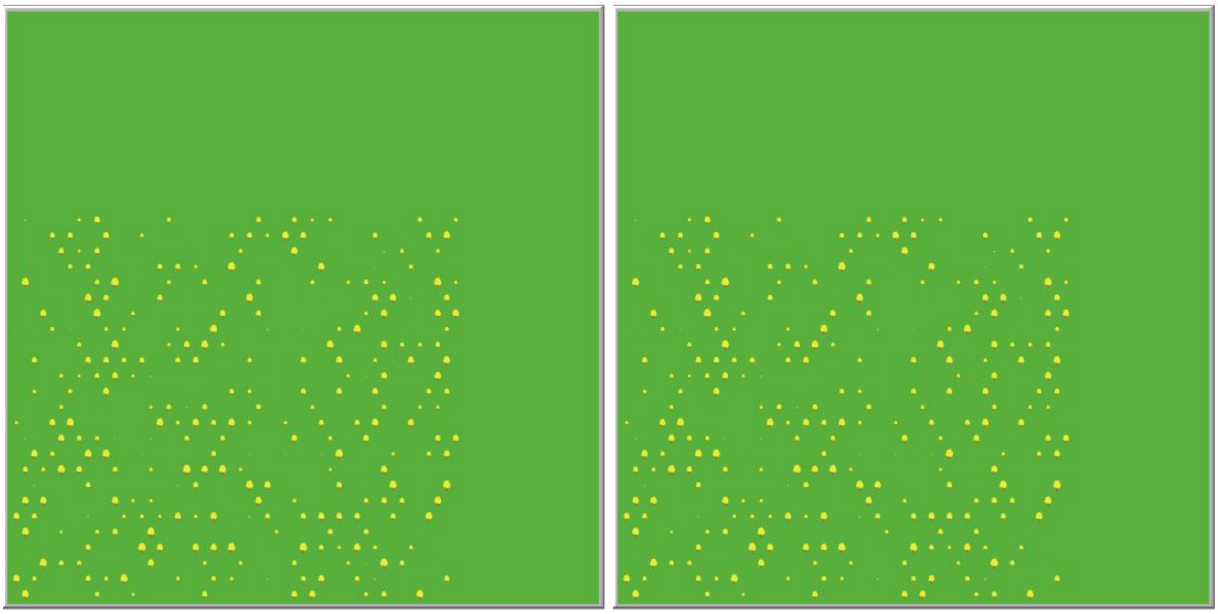


Figure 4.3: Simulation 2: Time series output of 70% Infection Proportion simulation using h=0.30, K_H = 0.2350, α = 0.6 , β = 0.9, γ = 0.8 , ψ = 20 with 25x25 plantation Global Control Strategy

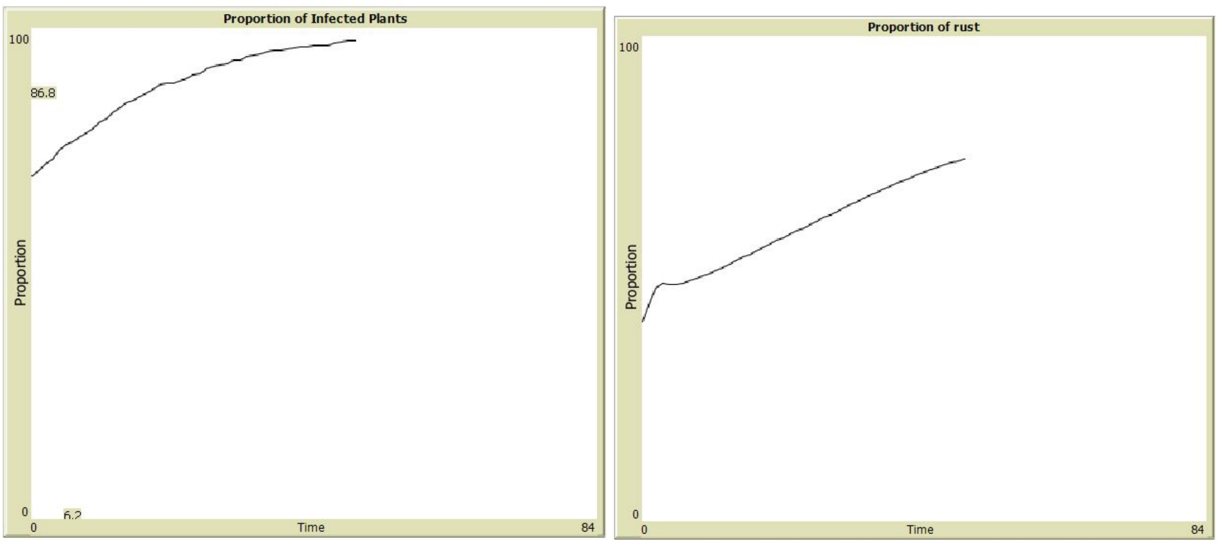


Figure 4.4: Proportions of infected and rust in Global Control Strategy in Simulation 2

In Simulation 2, we have a 70% Infection Proportion, 0.30 for rust growth, K_H = 0.2350 for the carrying capacity, 0.6 for the immigration rate, 0.9 for the emigration rate, 0.8 for the bacteria conversion rate and a 20 proportionality constant in a 25x25 grid plantation. Global Control Strategy is used here. The simulation has a starting high Infection proportion, the lower bound K_H,medium immigration rate, high emigration rate, high, and a medium proportion constant value. The results is a steady growth of the rust disease. The bacteria fails to suppress the epidemic and it only prevents any plant to have a high infection level, homogenize every plant with the same infection level. Continuing beyond the initial run, This results the plantation to be fully infected with the same infection level.

4.3 Simulation 3

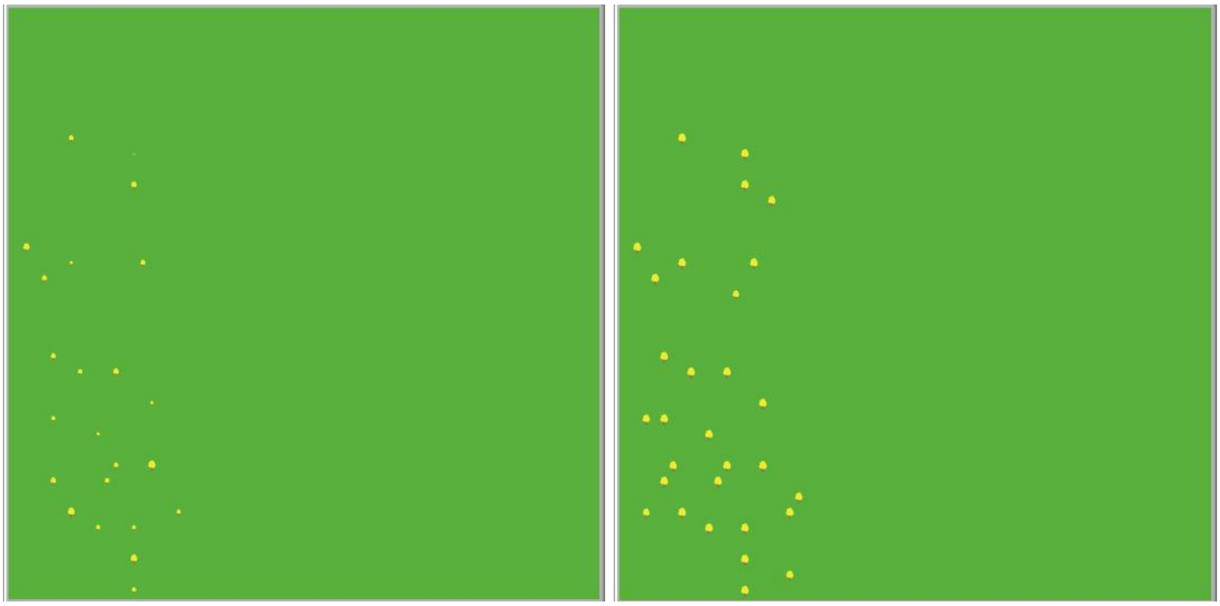


Figure 4.5: Simulation 3: Time series output of 10% Infection Proportion simulation using h=0.10, K_H = 0.2350, α=0.1 , β=0.1, γ=0.1 , ψ=10 with 30x30 plantation Global Control Strategy

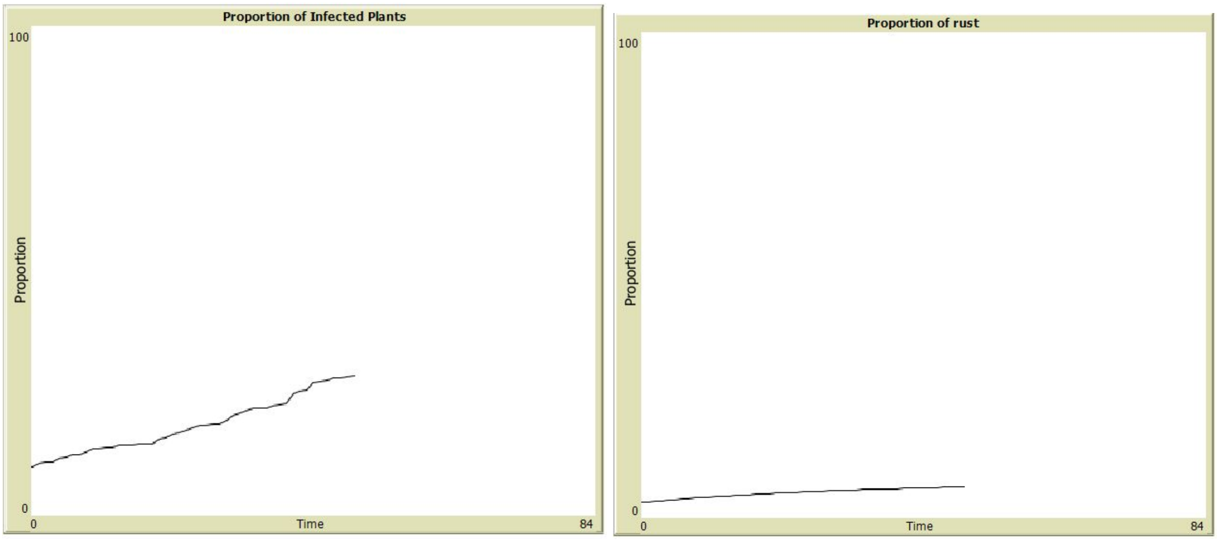


Figure 4.6: Proportions of infected and rust in Global Control Strategy in Simulation 3

In Simulation 3, we have a 10% Infection proportion, 0.10 rust spore growth rate, 0.1 for immigration and emigration rust spore rate ,0.1 for bacteria conversion rate and a 10 proportionality constant in a 30x30 plantation. Global Control Strategy is used here as well. The results are that the bacteria suppressing the coffee rust disease from spreading throughout the plantation. The initial infected plants failed to infect other plant to prolong the disease, instead they become fully infected to themselves only while other plants are healthy. This result happen likely due to the low growth rate and low immigration rate and emigration rate of rust spore. As these parameters are contributing for the coffee rust disease epidemic's existence and persistence against the fungicide. The lower their values,the lower the coffee rust growth, spread,and resistance. This simulation portrays that as even if the bacteria conversion rate is at the lowest value, the disease epidemic fails to spread out within 48 hours due to the said parameters having low values. This the first case that the bacteria is an effective fungicide against the Coffee Rust disease.

4.4 Simulation 4

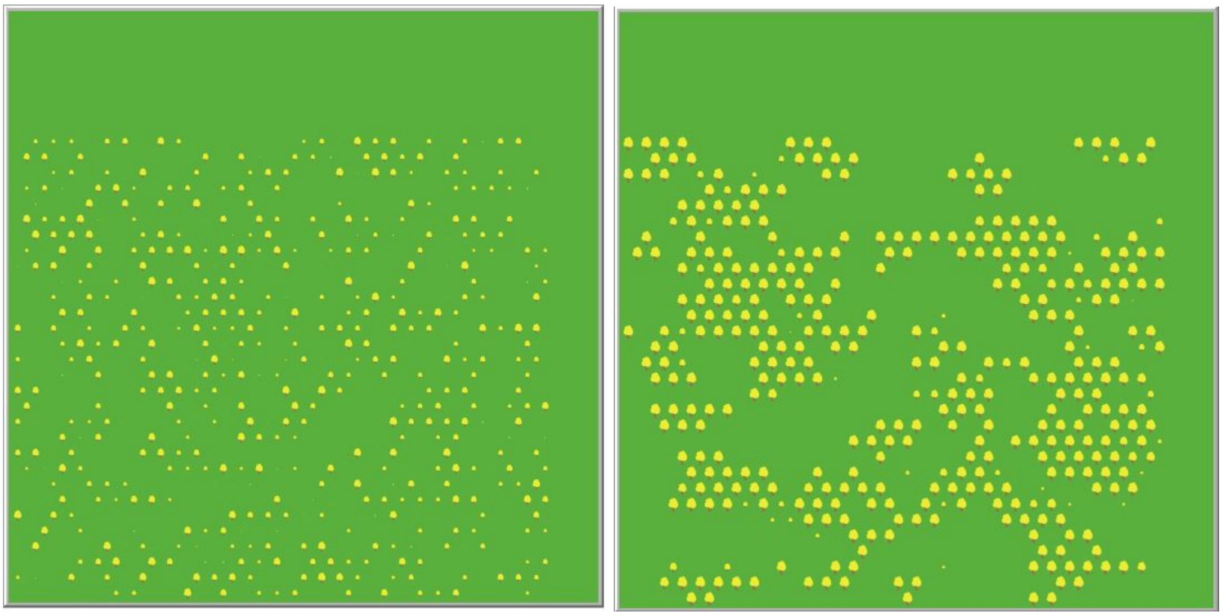


Figure 4.7: Simulation 4: Time series output of 80% Infection Proportion simulation using h = 0.15, K_H = 0.3000, $\alpha = 0.3$, $\beta = 0.7$, $\gamma = 0.7$, $\psi = 30$ with 30x30 plantation Global Control Strategy

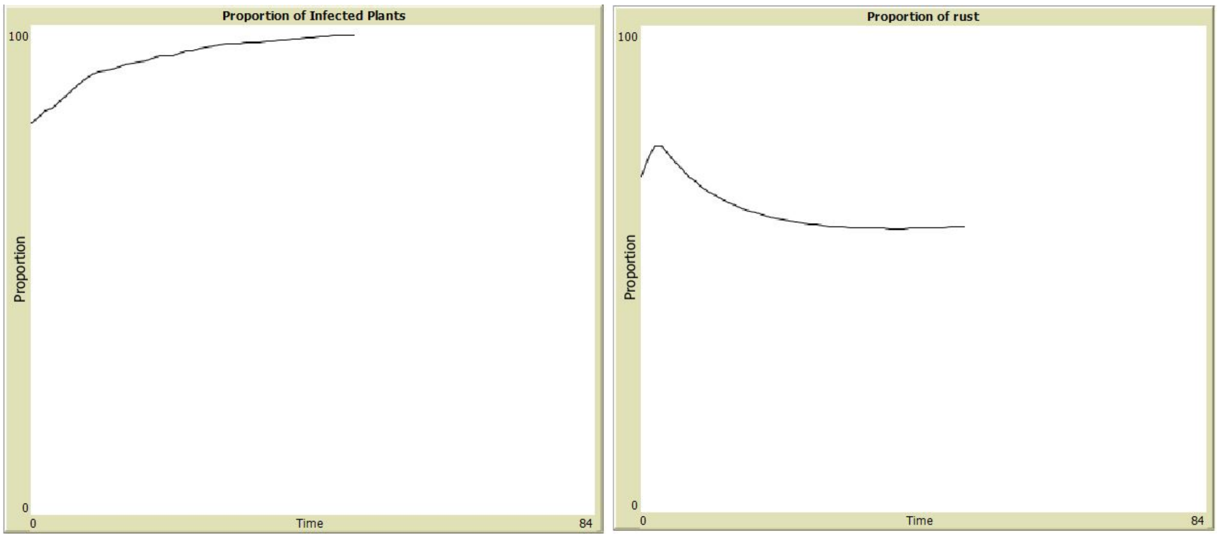


Figure 4.8: Proportions of infected and rust in Global Control Strategy in Simulation 4

For Simulation 4, we have a 80% Infection proportion, 0.15 rust spore growth rate, 0.7 for immigration and emigration rust spore rate, K_H=0.3000 for the carrying capacity, and a 30 proportionality constant in a 30x30 plantation. The simulation result is an unusual phenomenon. The bacteria seemingly able to suppress the rust for a while, evidently on the proportion of the rust graph. But after the 24 hour mark, it fails to suppressing the rust spore growth and the disease spread out. From the graphs results, it looks that plants with low infection level become healthy while plants with high infection continues to grow until they are 100% fully infection level. These plants infect they neighboring plants as well after that and will have the same infection level. In conclusion, the bacteria is able to suppress rust spore spread to only low infection level plants but fails on high infection level plants on the other hand. Further continuing the simulation run will result all plants to have 100% infection level.

4.5 Simulation 5

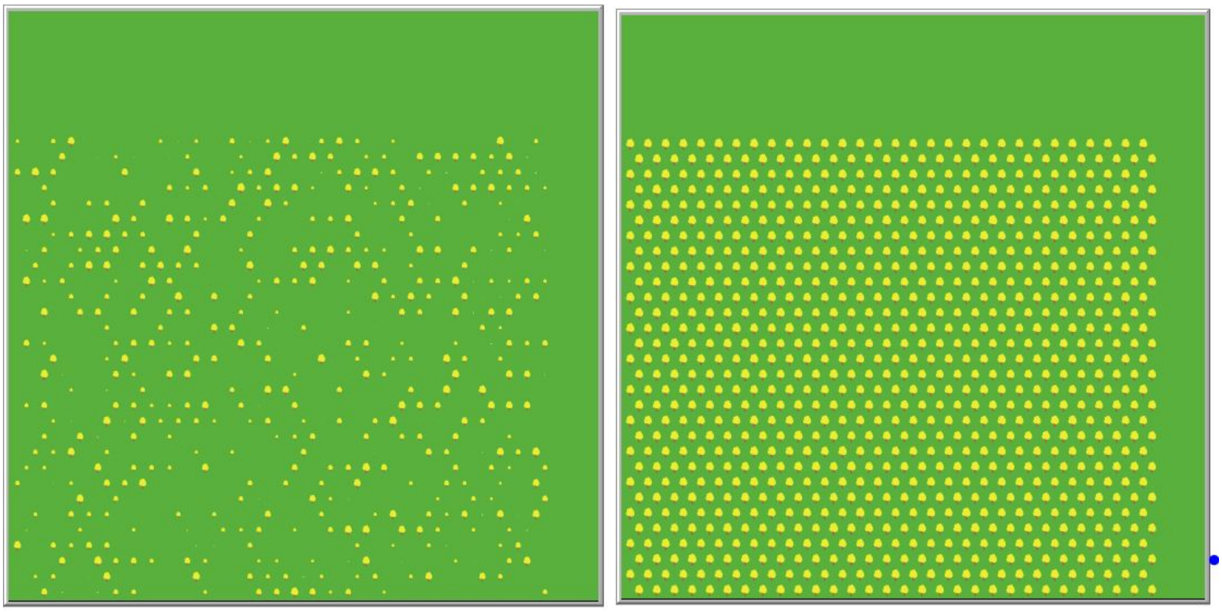
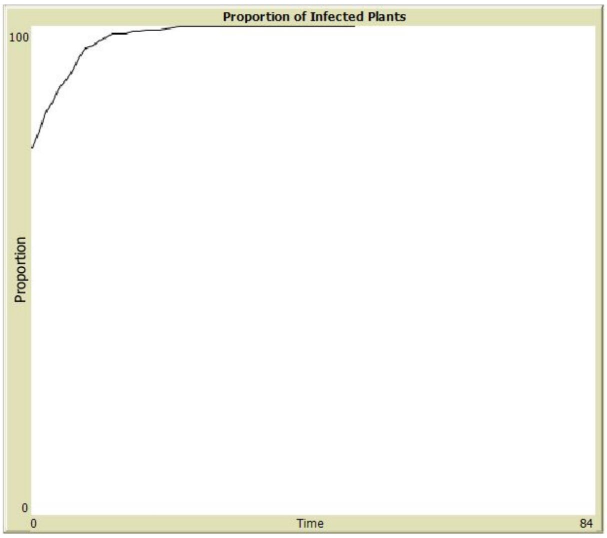


Figure 4.9: Simulation 5: Time series output of 75% Infection Proportion simulation using h=0.75, K_H = 0.3000, ρ = 0.9, δ = 0.8 α = 1 , β=1, γ=1 , ψ=50 with 30x30 plantation with Local Control Strategy



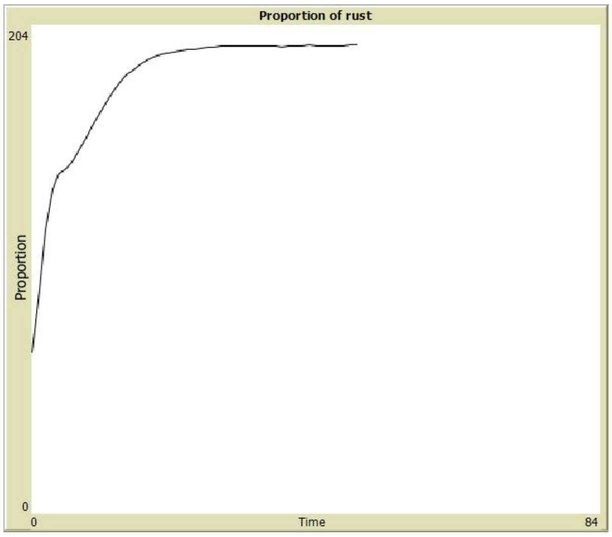


Figure 4.10: Proportions of infected and rust in Global Control Strategy in Simulation 5

For Simulation 5, a 75% infection proportion, 0.9 for main irrigation , 0.8 for second irrigation rate, 0.75 for growth rate, 1 for immigration, emigration and bacteria conversion rate, $K_H = 0.3000$ for the carrying capacity,and a 50 proportion constant in a 30x30 grid plantation. This time the control strategy chosen is Local Control Strategy. The resulting phenomenon is similar with Simulation 4, where the low-infection-level plants dies out while the high-infection-level plants increasing their infection level and infecting neighboring plants. The epidemic disease will able to fully infect all plants in the plantation in just 48 hours.The difference between Simulation 4 and 5 is the proportion of rust graph. Simulation 4's graph has a decreasing parabola shape with a sharp decreasing turn at the early simulation hours. Simulation 5 has an increasing parabola shape with a slight bump at the early simulation hours as well. It should be noted that further simulation of Simulation 4 will result it's graph rising and reaching the same peak proportion of rust in a very long time like about 120 hours after the initial simulation.

4.6 Simulation 6

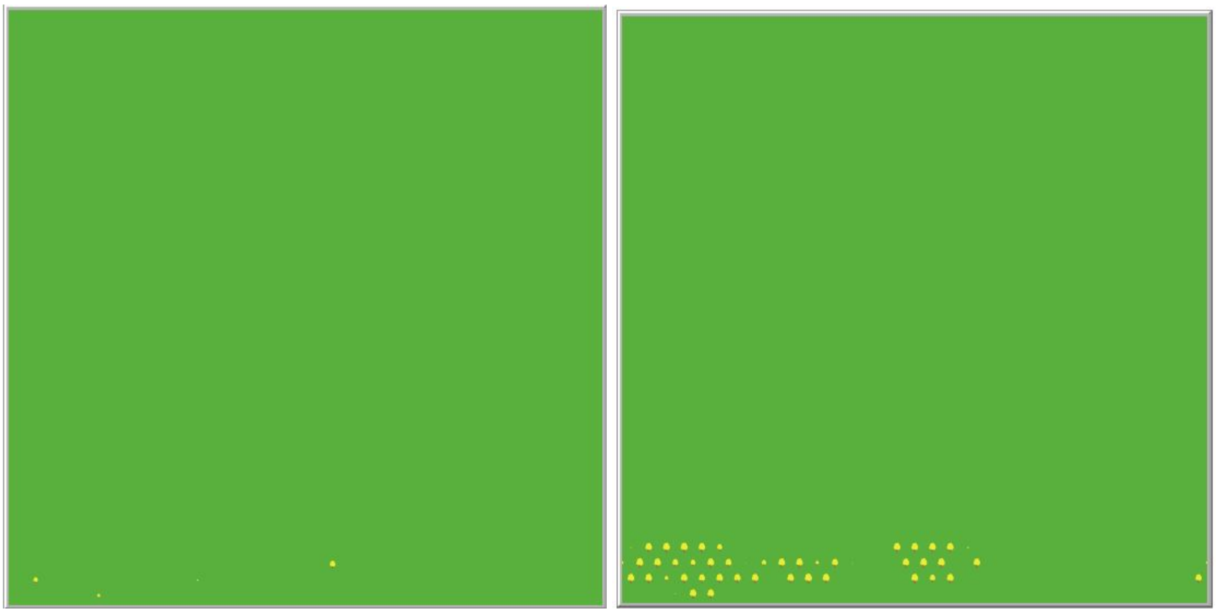


Figure 4.11: Simulation 6: Time series output of 4% Infection Proportion simulation using h=0.40, K_H=0.2350, ρ = 0.4, δ = 0.3, α=0.8 , β=0.8, γ=0.4 , ψ=40 with 40x40 plantation Local Control Strategy

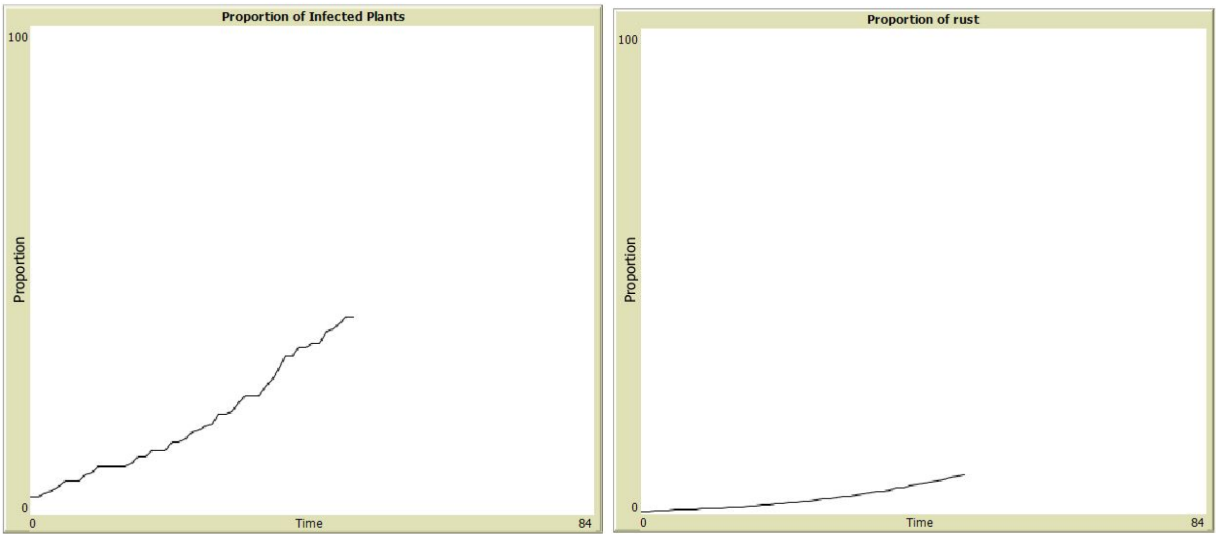


Figure 4.12: Proportions of infected and rust in Local Control Strategy in Simulation 6

In Simulation 6, we have a 4% Infection proportion, 0.40 rust spore growth rate , 0.4 for main irrigation rate, 0.3 for second irrigation rate, 0.8 for both immigration and emigration rust spore rate, 0.4 for bacteria conversion rate, $K_H = 0.2350$ for the carrying capacity, and a 40 proportionality constant in a 40x40 plantation. Local Control strategy is used here. The resulting run is a slow growth of the rust spore. It looks that the bacteria fails to suppress the rust disease. This happens due to high immigration and emigration rust spore of the rust disease and relative low bacteria conversion rate. This leads to the epidemic to slowly spreading throughout the plantation and infecting many plants. Further simulation beyond the 48 hour mark will result a fully infected plantation.

4.7 Simulation 7

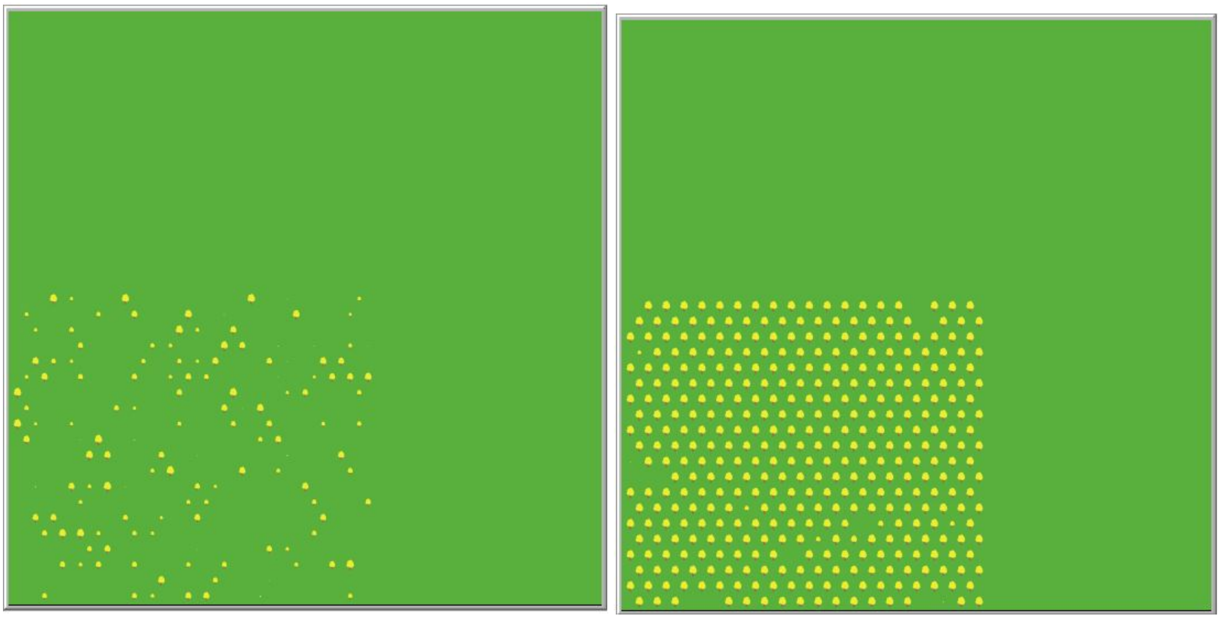


Figure 4.13: Simulation 7: Time series output of 50% Infection Proportion simulation using h = 0.55, $K_H = 0.2350$, $\alpha = 0.5$, $\beta = 0.5$, $\gamma = 0.5$, $\psi = 25$ in with 20x20 plantation Global Control Strategy

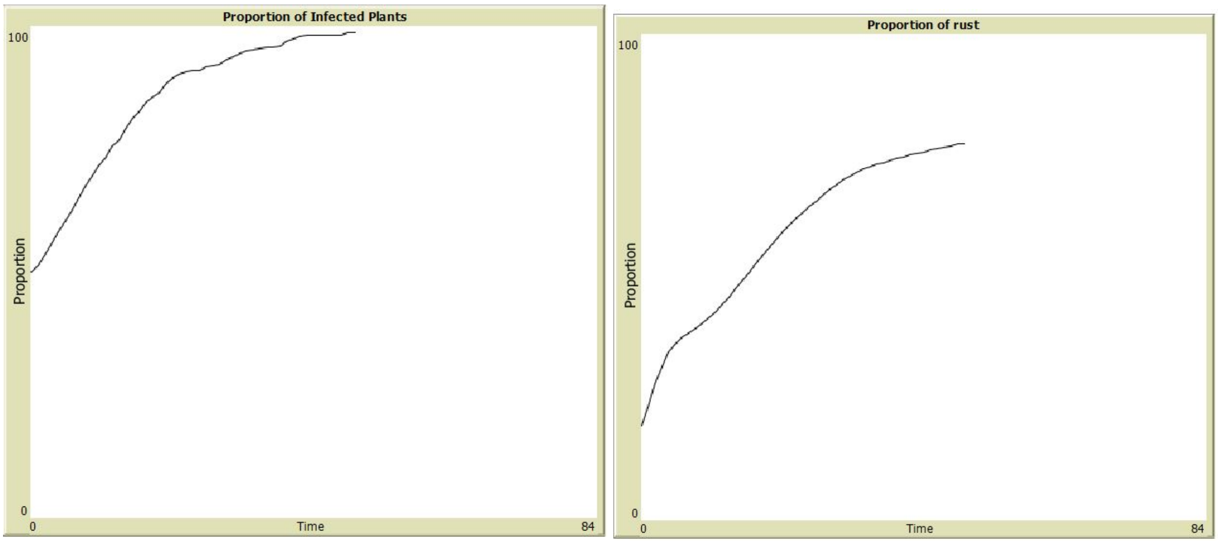


Figure 4.14: Proportions of infected and rust in Global Control Strategy in Simulation 7

In Simulation 7, we have a 50% Infection proportion, 0.55 rust spore growth rate, 0.5 for both immigration and emigration rust spore rate, 0.5 for bacteria conversion rate, $K_H = 0.2350$ for the carrying capacity and a 25 proportionality constant in a 20x20 plantation with Global Control Strategy selected. Most of these parameters have a value of '5'. The simulation run shows the bacteria tremendously failed to inhibit the coffee rust epidemic. At the start, the infection proportion is perfectly be half to the healthy plants. The end result is a nearly infected plant plantation with only few healthy plants. Continuing the simulation from initial run will result a fully infected plantation in just matter of seconds in real life.

4.8 Simulation 8

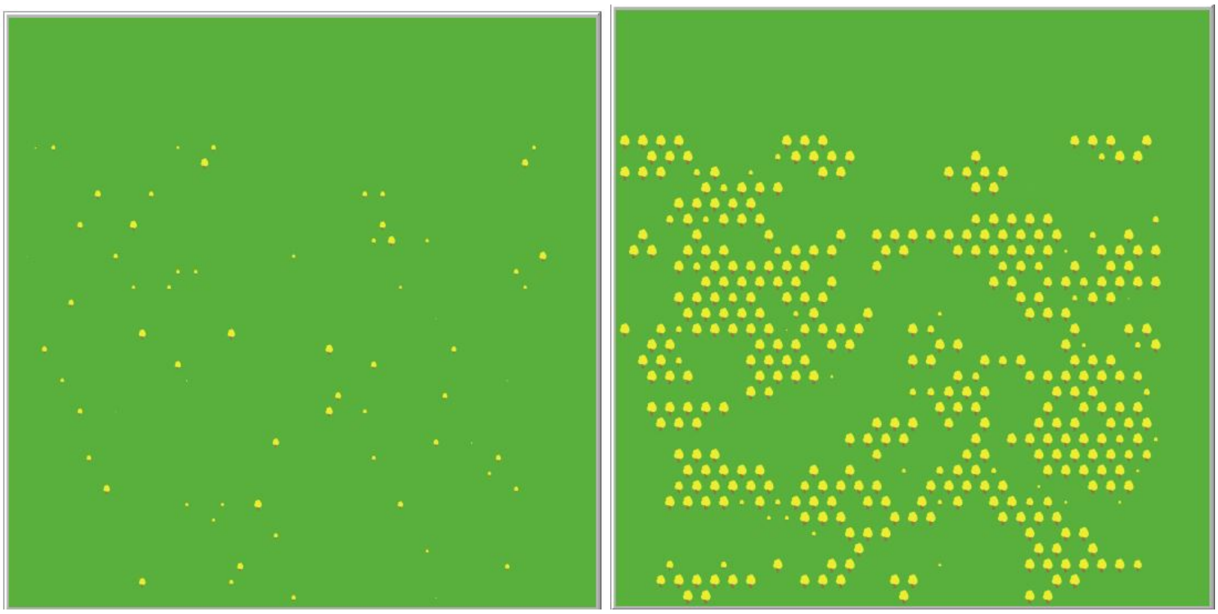


Figure 4.15: Simulation 8: Time series output of 80% Infection Proportion simulation using h = 0.55, K_H = 0.3000, α = 0.9 , β = 0.7, γ = 0.7 , ψ = 10 in with 30x30 plantation Global Control Strategy

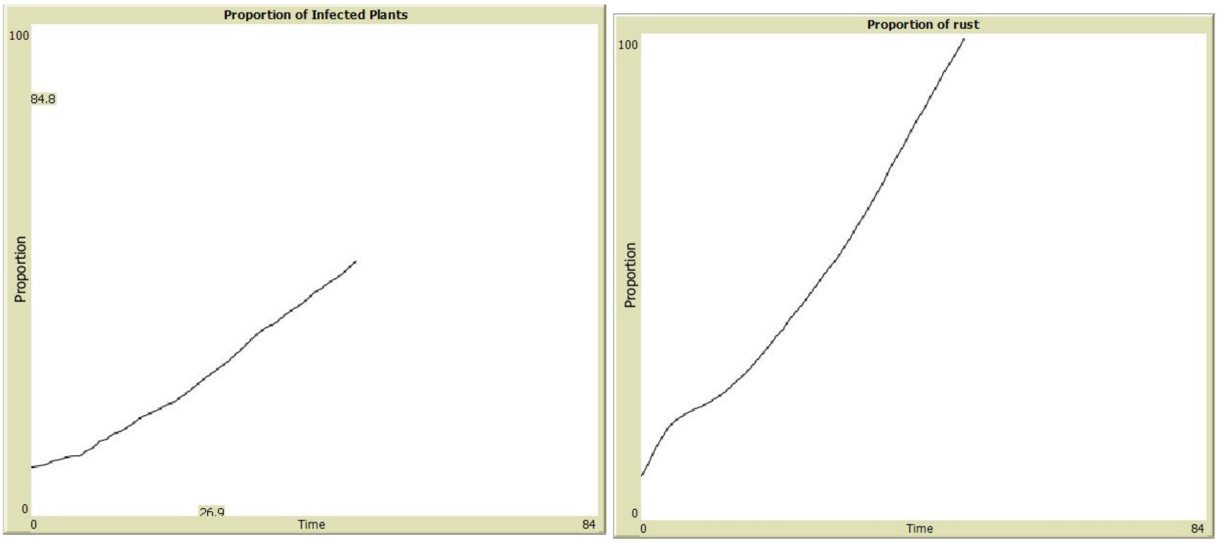


Figure 4.16: Proportions of infected and rust in Global Control Strategy in Simulation 8

For Simulation 8, we have a 80% Infection proportion, 0.55 rust spore growth rate, 0.9 for immigration rate, 0.7 for emigration rust spore rate, 0.7 for the conversion rate, K_H=0.3000 for the carrying capacity, and a 10 proportionality constant in a 30x30 plantation with Global Control Strategy being selected. The results show bacteria unable to suppress the coffee rust disease. The coffee rust proportion graph shows a linear growth of the disease. There is a slight bump by 5-10 hour marks, indicating that the bacteria has somehow slow down the rust disease. But the Coffee Rust will still grow and overcome the bacteria inhibition effect. Further simulation results in a fully infected plantation in just hours.

In the following two simulations, both Global and Local Control Strategies will be applied separately, while maintaining consistent parameter values within each strategy.

4.9 Simulation 9(Global Control Strategy)

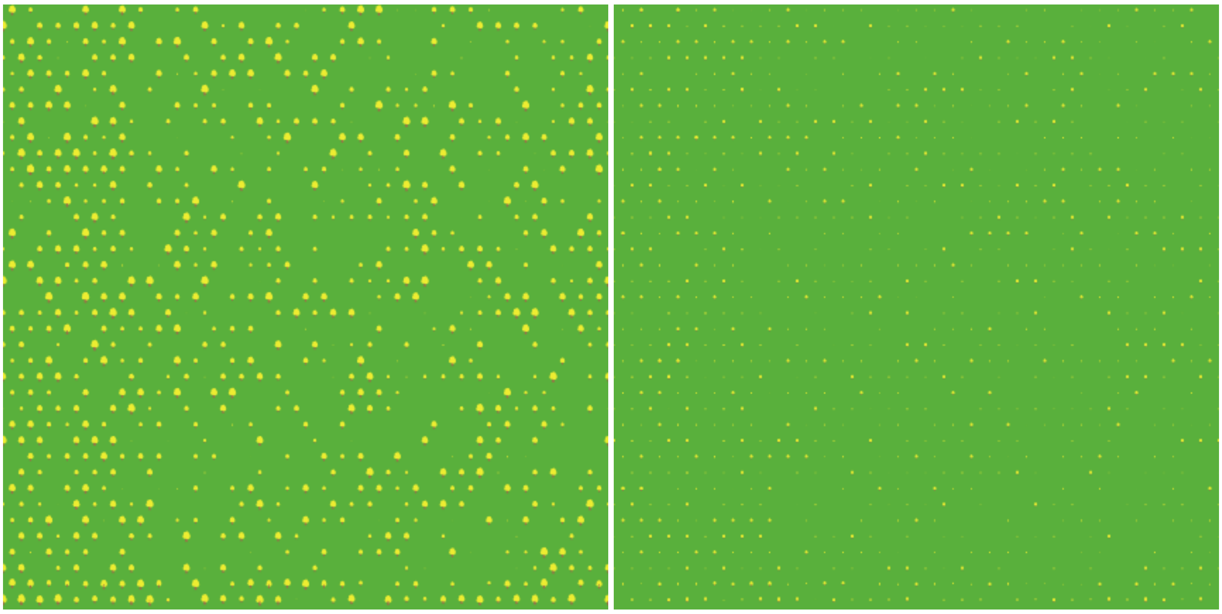


Figure 4.17: Simulation 9 (Global Control Strategy): Time series output of 80% Infection Proportion simulation using h = 0.15, K_H = 0.2350, α = 0.9 , β = 0.1, γ = 0.9 , ψ = 5 in Global Control Strategy

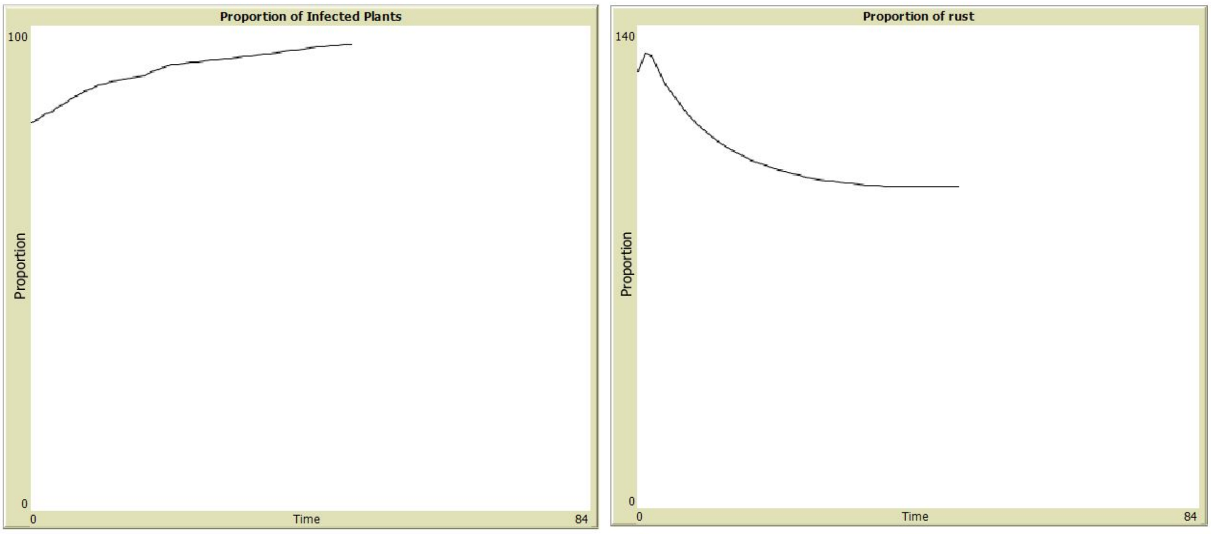


Figure 4.18: Proportions of infected and rust in Global Control Strategy in Simulation 9 (Global Control Strategy)

Now for special Simulation 9 in Global Control Strategy, we have a 80% Infection proportion, 0.15 rust spore growth rate, K_H=0.2350 for the carrying capacity, 0.9 for immigration rate, 0.1 for emigration rust spore rate, 0.9 for the conversion rate and a 5 proportionality constant in a 40x40 plantation. This implies a high infection starting Infection proportion, low rust spore growth rate, high immigration rust rate, low emigration rust rate, high bacteria conversion rate, and a very low proportion constant. The initial simulation run results is bacteria able to suppress the disease but the disease will overcome it after the 48 hour mark. This cause is due to high immigration rust rate as it makes the disease to prolong their existence by infecting plants. This leads every plant to have the same infection level. Bacteria fails in preventing the epidemic disease spreading in the plantation. It only manages to make the infection level of every plant to be uniformly low level. Coffee Rust disease persist the bacteria's inhibition effect.

4.10 Simulation 9 (Local Control Strategy)

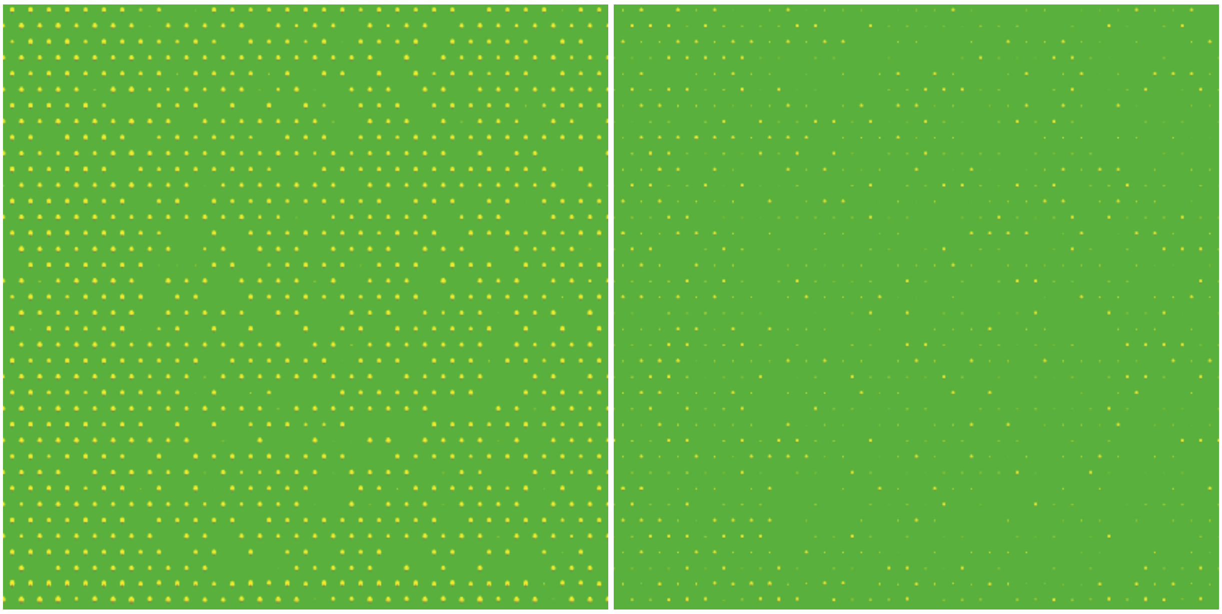


Figure 4.19: Simulation 9 (Local Control Strategy) Time series output of 80% Infection Proportion simulation using h = 0.15, K_H = 0.2350, $\rho = 1$, $\delta = 0.9$, $\alpha = 0.9$, $\beta = 0.1$, $\gamma = 0.9$, $\psi = 5$ in Local Control Strategy

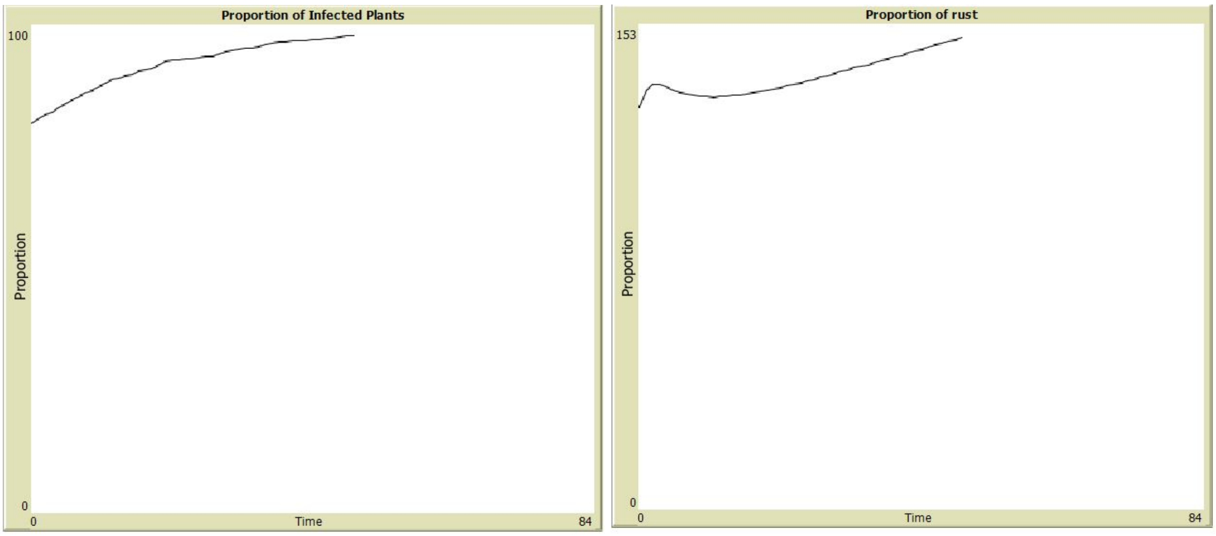


Figure 4.20: Proportions of infected and rust in Local Control Strategy in Simulation 9 (Local Control Strategy)

In the respective Local Control Strategy of Simulation 9, the change is in the irrigation rate. 1 on the main irrigation rate and 0.9 on the second irrigation rate, it is almost the same phenomenon but with slightly inferior results. The infection level on most plants are much bigger compared in the Global Control Strategy. The differences sizes of the yellow circles of Global and Local Control Strategies show these phenomenon. With these results from the 2 Control Strategies, Global Control Strategy is the more effective control strategy.

4.11 Simulation 10(Global Control Strategy)

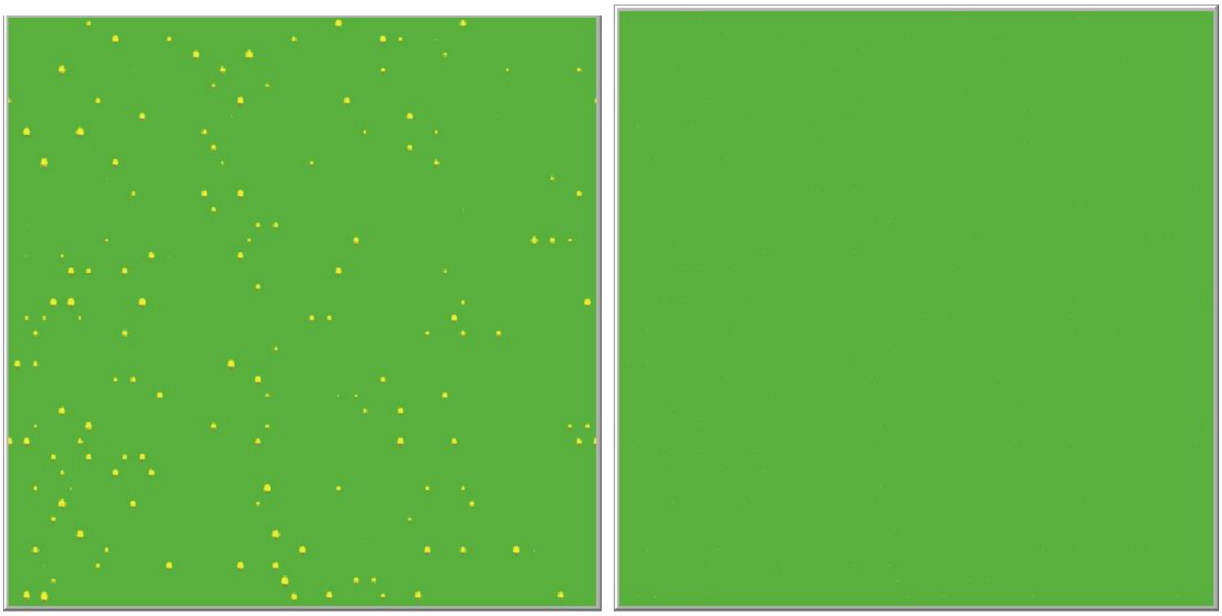


Figure 4.21: Simulation 10 (Global Control Strategy): Time series output of 15% Infection Proportion simulation using h = 0.15, $K_H = 0.2350$, $\alpha = 0.9$, $\beta = 0.1$, $\gamma = 0.9$, $\psi = 5$ in Global Control Strategy

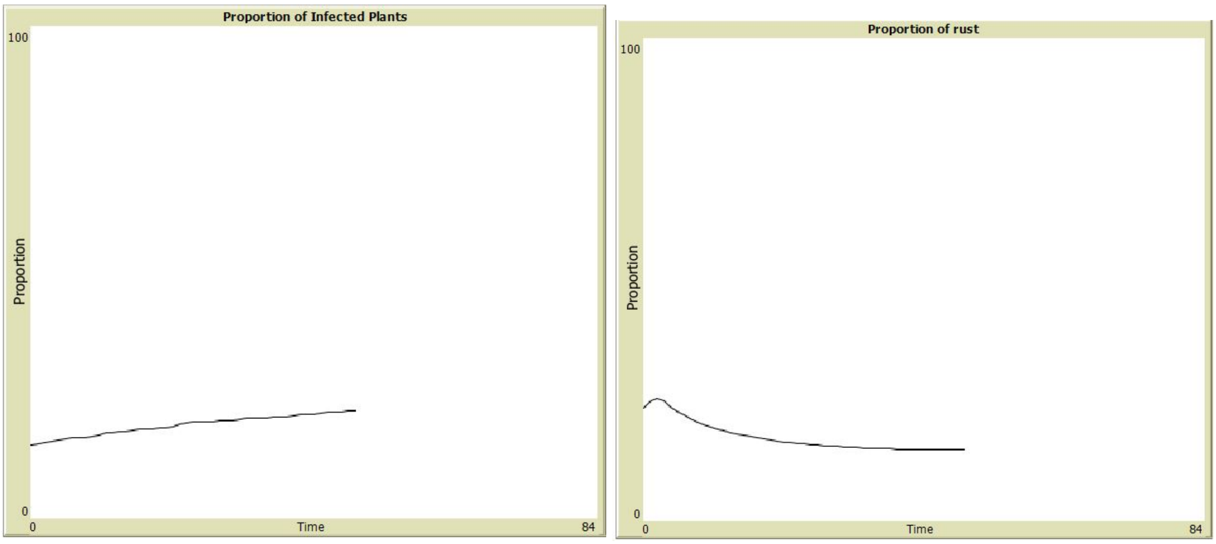


Figure 4.22: Proportions of infected and rust in Global Control Strategy in Simulation 10 (Global Control Strategy)

Now for special Simulation 10 in Global Control Strategy, we have a 15% Infection proportion, 0.15 rust spore growth rate, 0.9 for immigration rate, 0.1 for emigration rust spore rate, 0.9 for the conversion rate, $K_H = 0.2350$ for the carrying capacity and a 5 proportionality constant in a 40x40 plantation. This simulation graphs show that coffee rust disease will die out within 48 hours. This means that the bacteria suppresses the disease within those hours. The epidemic disease is unable to persist due to failing to infect neighboring plants. The explanation is due to high conversion rate, low initial Infection proportion, low rust spore emigration and low rust spore growth rate. The initial state of the disease is low level throughout the plantation while the bacteria suppressing effect is high. Combining this with the low emigration rate, the bacteria prevents the spread of Coffee rust spores in infecting new Coffee plants. The Coffee Rust disease dies out eventually due to low growth rate.

4.12 Simulation 10 (Local Control Strategy)

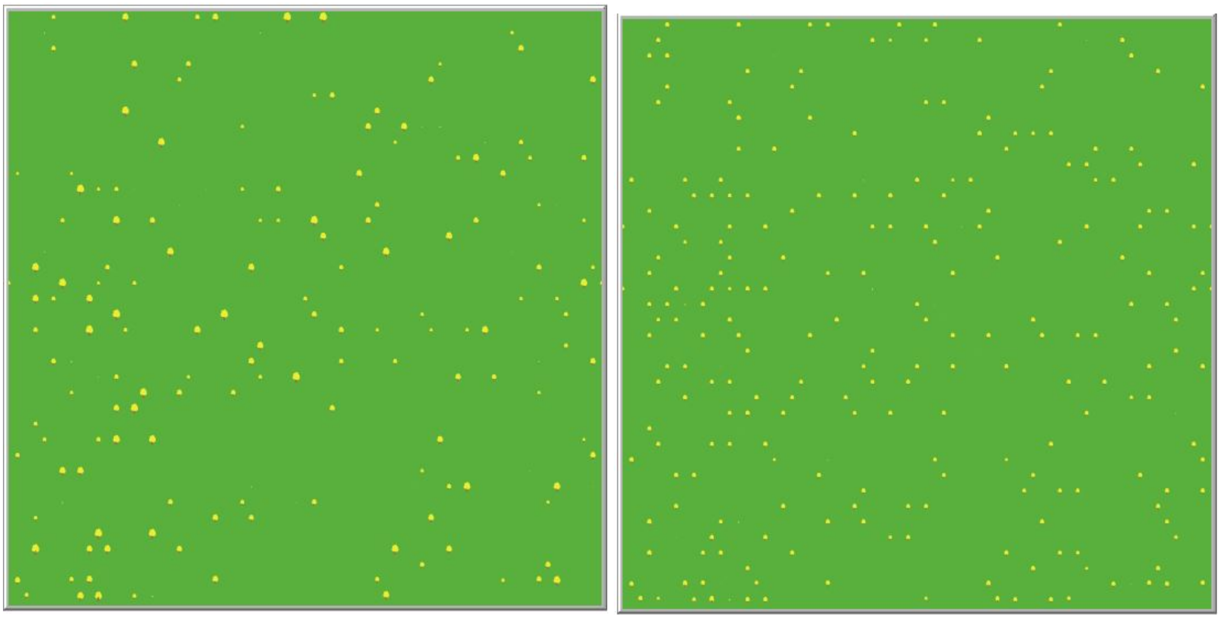


Figure 4.23: Simulation 10 (Local Control Strategy): Time series output of 15% Infection Proportion simulation using h = 0.15, $K_H = 0.2350$, $\alpha = 0.9$, $\beta = 0.1$, $\gamma = 0.9$, $\psi = 5$ in Local Control Strategy

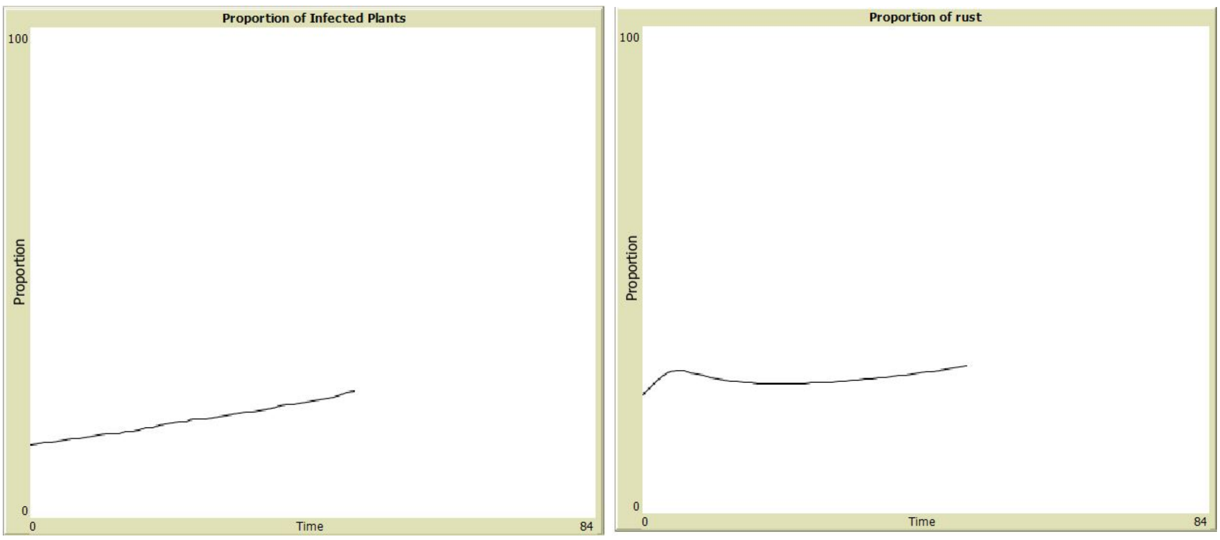


Figure 4.24: Proportions of infected and rust in Local Control Strategy in Simulation 10 (Local Control Strategy)

In the respective Local Control Strategy of Simulation 10, the change is in the irrigation rate. 1 on the main irrigation rate and 0.9 on the second irrigation rate. There is a stark difference between the two Control Strategies evidently on the 48 hour mark of the plantation. The bacteria has less suppressing efficiency compared in Global Control Strategy.The epidemic disease is still able to spread out in the plantation although in low-infection-levels per plant. The rust disease persist the suppression effect. With these results from the 2 Control Strategies, Global Control Strategy is the more effective control strategy.

Chapter 5

Conclusion and Recommendations

From these results, we have successfully built a simulation model of the hexagonal plantation of *Kapeng Barako* with aide of the bacteria *Bacillus subtilis* as the biological control agent. The results shown that *Bacillus subtilis* can be a replacement of *Bacillus thuringiensis* or other copper pesticides. However, it is noted that this bacteria can be an effective fungicide depending on the parameters. These parameters have been studied in their effect in the simulation. The most important factor is the bacteria conversion rate. As this parameter dictates on how efficient is the suppressing effect of the bacteria. In real life application, this efficiency is not fix and can change overtime. Low rust growth rate, low immigration emigration rust rates and low proportion constant are parameters that also contribute in making *Bacillus subtilis* an effective fungicide.

In terms of differences in performance of the two Control Strategies, Both Global and Local Control Strategies have instances to be an effective counter strategies against the disease epidemic. But the winning strategy is the Global Control Strategy. As it is the more effective control strategy in suppressing the epidemic disease than Local Control. It is noted that Local Control Strategy has it's strength. If cost of the bacteria fungicide is considered, then Local Control is the chosen strategy. As it still is an efficient control strategy while also cut cost for the fungicide.

Overall, the objectives of this study have been achieve: a simulation model, cases where the bacteria inhibits the coffee rust growth, the analyzed contributing parameters,and the examination of the 2 control strategies efficiency.

For future researchers who wants to use or improve this model, here are some suggested ideas. There is a great need for improvements functions on the model in order be more realistically operative. Things like Wind, insects role in the ecosystem, Topological factors, temperature factors and human action effects. Change in plantation is also recommended. The suggested layout is the Contour shape. As this layout is commonly be used by coffee farmers especially in the Philippines. Change in coffee species and the antagonistic disease may do so as long as the parameters values change to fit with the modifications. This model can be use to other plants such as rice, fruit trees, Corn, etc.

Bibliography

[1] Bacillus subtilis or b. subtilis hay bacillus or grass bacillus. Fengchen Group.

[2] Coffee technoguide: Technical guidebook on coffee plant varieties, cultivation, harvesting, processing, and branding. n.d.

[3] Abdelwahab, S. I., Taha, M. M. E., Jerah, A. A., Aljahdali, I. A., Oraibi, B., Alfaifi, H. A., Abdullah, S. M., Alzahrani, A. H., Oraibi, O., Babiker, Y., and Farasani, A. Coffee arabica research (1932–2023): Performance, thematic evolution and mapping, global landscape, and emerging trends. *Heliyon 10* (2024), e36137.

[4] Amaquin, J. M. B. Kapeng barako in a tea bag. https://www.scribd.com/document/564060560/Kapeng-Barako-Research-Paper-BSBA-3201-Amaquin-Jaqueline-Mae-B, 2021. BSBA-3201 Marketing Research Paper, Batangas State University.

[5] Arroyo-Esquivel, J., Sanchez, F., and Barboza, L. Infection model for analyzing biological control of coffee rust using bacterial anti-fungal compounds. *Mathematical Biosciences 307* (2019), 13–24.

[6] Baite, T. N., Mandal, B., and Purkait, M. K. Extraction of coffee and tea. In *Extraction Processes in the Food Industry*, M. K. Purkait, Ed. Elsevier, 2024, ch. 9, pp. 247–274.

[7] Borriss, R., Danchin, A., Harwood, C. R., et al. Bacillus subtilis, the model gram-positive bacterium: 20 years of annotation refinement. *Microbial Biotechnology 11*, 1 (2018), 3–17.

[8] Gibson, M., and Newsham, P. Chapter 18 - tea and coffee. In *Food Science and the Culinary Arts*, M. Gibson and P. Newsham, Eds. Academic Press, 2018, pp. 353–372.

[9] Haddad, F., Maffia, L., Mizubuti, E., and Teixeira, H. Biological control of coffee rust by antagonistic bacteria under field conditions in brazil. *Biological Control 49*, 1 (2009), 114–119.

[10] Haque, M. A., and Sakimin, S. Z. Planting arrangement and effects of planting density on tropical fruit crops—a review. *Horticulturae 8*, 6 (2022), 485.

[11] Holmberg, T. J. 4.2.5: Population models. https://bio.libretexts.org/@go/page/105424, 2024. Accessed: 2025-05-15.

[12] Knapp, S. Bacillus subtilis – the definitive guide. Biology Dictionary, 2020.

[13] Mahmud, S., Hassan, M., Moniruzzaman, M., Biswas, N., Rahman, M. M., and Haque, M. Study on the accumulation of copper from soil by shoots and roots of some selective plant species. *International Journal of Biosciences (IJB) 3* (06 2013), 68–75.

[14] Mangel, M., and Roitberg, B. D. Behavioral stabilization of host-parasite population dynamics. *Theoretical Population Biology 42*, 3 (1992), 308–320.

[15] Manson, S. Simulation. In *International Encyclopedia of Human Geography*, R. Kitchin and N. Thrift, Eds. Elsevier, Oxford, 2009, pp. 132–137.

[16] Nugroho, D., Basunanda, P., and Mw, S. Physical bean quality of arabica coffee (coffea arabica) at high and medium altitude. *Pelita Perkebunan 32*, 3 (2016), 177–194.

[17] Philippine Coffee Board Inc. Philippine coffee. `https://philcoffeeboard.com/philippine-coffee/`, 2025. Accessed: 2025-05-14.

[18] Poggere, G., Gasparin, A., Barbosa, J. Z., Melo, G. W., Corrêa, R. S., and Motta, A. C. V. Soil contamination by copper: Sources, ecological risks, and mitigation strategies in brazil. *Journal of Trace Elements and Minerals 4* (2023), 100059.

[19] Rayner, R. W. Germination and penetration studies on coffee rust (hemileia vastatrix b. & br.). *Annals of Applied Biology 49*, 3 (1961), 497–505.

[20] Schacht, R. Two models of population growth. *American Anthropologist 82* (10 2009), 782 – 798.

[21] Smith, J., and Brown, A. A study on algorithms. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (2020), pp. 123–134.

[22] Tumber-Dávila, S. J., Schenk, H. J., Du, E., and Jackson, R. B. Plant sizes and shapes above and belowground and their interactions with climate. *New Phytologist 235*, 3 (2022), 1032–1056.

[23] Vandermeer, J., and King, A. Consequential classes of resources: Subtle global bifurcation with dramatic ecological consequences in a simple population model. *Journal of Theoretical Biology 263* (2010), 237–241.

[24] Waller, J. Coffee rust—epidemiology and control. *Crop Protection 1*, 5 (1982), 385–404.

[25] Wangersky, P. J. Lotka-volterra population models. *Annual Review of Ecology and Systematics 9* (1978), 189–218.

[26] Wilensky, U. Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University.

[27] Yadav, U., Anand, V., Kumar, S., et al. Endophytic biofungicide bacillus subtilis (nbri-w9) reshapes the metabolic homeostasis disrupted by the chemical fungicide, propiconazole in tomato plants to provide sustainable immunity against non-target bacterial pathogens. *Environmental Pollution 343* (2024), 123144.

[28] Zhang, B., and DeAngelis, D. L. An overview of agent-based models in plant biology and ecology. *Annals of Botany 126*, 4 (2020), 539–557.

Appendix

Source Code

```
turtles-own [
  proportion          ;; Proportion of bacteria present
  proportion-temp     ;; Temporary proportion storage for calculations
  bacteria-size       ;; Size of bacteria individuals
]

to setup
  clear-all           ;; Clears the world and resets the model
  setup-globals       ;; Initializes global variables
  setup-patches       ;; Prepares the environment/patches
  setup-turtles       ;; Creates and initializes bacteria agents (turtles)
  reset-ticks         ;; Resets the tick counter to start the simulation
end

to setup-globals
  set growth-rate-bacteria 2      ;; Sets the bacteria growth rate to 2
  set carrying-cap-bacteria 7     ;; Maximum population capacity for bacteria is 7
  set death-rate 1.5              ;; Death rate of bacteria is 1.5

  set growth-rate-rust 0.15       ;; Rust growth rate is set to 0.15
  set carrying-cap-rust 0.284     ;; Maximum carrying capacity for rust is 0.284
  set immigration-rate 0.9 / 24   ;; Rate at which bacteria enter the system per unit time
  set emmigration-rate 0.1 / 24   ;; Rate at which bacteria leave the system per unit time
  set conversion-rate 0.9 / 24    ;; Rate at which bacteria convert to rust
  set prop-prob 20               ;; Probability related to bacteria proportion changes

  set step-size 1                ;; Defines the step size for movement
  set sigma 3                    ;; Standard deviation for movement calculations
  set stepH 3                     ;; Step size in the horizontal direction

  set distH 4                     ;; Horizontal movement distance factor

  set distD 5                ;; Diagonal movement distance factor (approximate sqrt(5))
end

to setup-patches
  ask patches [set pcolor green] ;; Sets all patches (environment) to green, representing the initial
       state of the environment
end

to setup-turtles
  let horizontal-spacing (world-width / column-number)  ;; Calculates the horizontal spacing between
      turtles based on world width and column count
  let vertical-spacing (world-height / row-number)      ;; Calculates the vertical spacing between
      turtles based on world height and row count
```

```
  let min-xpos (min-pxcor - 0.5 + horizontal-spacing / 2)  ;; Determines the minimum x-coordinate for
      turtle placement
  let min-ypos (min-pycor - 0.5 + vertical-spacing / 2)  ;; Determines the minimum y-coordinate for
      turtle placement


  set-default-shape turtles "tree" ;; Sets the default shape of turtles to a tree

  create-turtles(column-number * row-number) [
    let xcor-value who mod column-number
    let ycor-value floor (who / column-number)  ;; Multiple rows

    let offset 0
    if (ycor-value mod 2 = 0) [set offset 0.5]  ;; Shift alternate rows for hexagonal effect

    setxy (xcor-value + offset) (ycor-value * 0.87)  ;; Proper hexagonal spacing

  ]



   ask turtles [
    set color yellow  ;; Sets all turtles to yellow
    set bacteria-size 1  ;; Initializes the size of bacteria to 1
  ]

  ask n-of (floor((count turtles) * infected-proportion / 100)) turtles [  ;; Randomly selects a
      percentage of turtles to be infected
    set proportion random-float 0.2  ;; Assigns them a random proportion value between 0 and 0.2
  ]

  ask turtles [
    set size proportion * 3  ;; Scales the turtle size based on its proportion value
    set proportion-temp proportion  ;; Stores the initial proportion value in a temporary variable
  ]
end

to-report report-proportion
  let S 0  ;; Initializes variable S to store the sum of proportions
  ask turtles [set S (S + ceiling(proportion - 0.00005))]  ;; Sums up the proportion values of all
      turtles, rounding slightly
  report S  ;; Returns the total proportion value
end

to go
  ask turtles [set size proportion * 3]  ;; Updates the size of each turtle based on its proportion
      value
  grow-rust  ;; Calls the procedure to simulate rust growth
  grow-bacteria  ;; Calls the procedure to simulate bacteria growth
  tick  ;; Advances the simulation by one time step

  if ticks = 48 [stop]  ;; Stops the simulation after 48 ticks
end

to grow-rust
   ask turtles [
```

```
    set proportion (proportion
      + proportion-temp * growth-rate-rust * (1 - proportion-temp / carrying-cap-rust)  ;; Logistic
          growth model for rust
      - emmigration-rate * proportion-temp  ;; Reduces rust proportion due to emigration
      - conversion-rate * bacteria-size * proportion-temp)  ;; Rust converts bacteria, reducing its
          own proportion
  ]

  disperse-rust  ;; Calls a function that likely spreads rust across turtles or patches

  ask turtles[
    set proportion max list 0 proportion  ;; Ensures that the proportion does not go below 0
    set proportion-temp proportion  ;; Updates the temporary proportion variable for the next
        calculation step
  ]
end

to disperse-rust
  let tau-1-0 0  ;; Neighboring turtle offset: left
  let tau+1-0 0  ;; Neighboring turtle offset: right
  let tau-0-1 0  ;; Neighboring turtle offset: below
  let tau-0+1 0  ;; Neighboring turtle offset: above
  let tau-1-1 0  ;; Neighboring turtle offset: bottom-left diagonal
  let tau-1+1 0  ;; Neighboring turtle offset: top-left diagonal
  let tau+1-1 0  ;; Neighboring turtle offset: bottom-right diagonal
  let tau+1+1 0  ;; Neighboring turtle offset: top-right diagonal

  let num 0      ;; Variable to store the number of possible dispersal directions
  let H-prob 0   ;; Horizontal dispersion probability
  let V-prob 0   ;; Vertical dispersion probability
  let D-prob 0   ;; Diagonal dispersion probability

  ask turtles[
    ;; Reset dispersion probabilities for each turtle
    set tau-1-0 0
    set tau+1-0 0
    set tau-0-1 0
    set tau-0+1 0
    set tau-1-1 0
    set tau-1+1 0
    set tau+1-1 0
    set tau+1+1 0

    ;; Check if turtle is on the top row
    ifelse who <  column-number[
      ;; If at the top-left corner
      ifelse who = 0[
        set H-prob dispProbability (1)(distH)  ;; Compute probability of moving right

        set D-prob dispProbability (column-number + 1)(distD)  ;; Compute probability of moving
            diagonally down-right

        if (random-float 1) <= D-prob [set tau-1+1 1]  ;; Move diagonally down-right
        if (random-float 1) <= H-prob [set tau-0+1 1]  ;; Move right
```

```
      ;; Update proportion based on immigration from neighboring turtles
      set proportion (proportion + immigration-rate * (tau-1-0 * [proportion-temp] of turtle column
          -number +
        tau-0+1 * [proportion-temp] of turtle 1 +
        tau-1+1 * [proportion-temp] of turtle (column-number + 1)))
      ]
       ;; If at the top-right corner
         [ifelse who = (column-number - 1)[
        set H-prob dispProbability (column-number - 2)(distH)
        if (random-float 1) <= H-prob [set tau-0-1 1]    ;; Move left

        set D-prob dispProbability (2 * column-number - 2)(distD)
        if (random-float 1) <= D-prob [set tau-1-1 1]    ;; Move diagonally down-left

        set proportion (proportion + immigration-rate * (tau-1-0 * [proportion-temp] of turtle (2 *
             column-number - 1) +
          tau-0-1 * [proportion-temp] of turtle (column-number - 2) + tau-1-1 * [proportion-temp]
             of turtle (2 * column-number - 2)))
      ][;; If in the top row but not in a corner
        set H-prob dispProbability (who - 1)(distH)
        if (random-float 1) <= H-prob [set tau-0-1 1]  ;; Move left
        set H-prob dispProbability (who + 1)(distH)
        if (random-float 1) <= H-prob [set tau-0+1 1]  ;; Move right

        set D-prob dispProbability (column-number + 1)(distD)
        if (random-float 1) <= D-prob [set tau-1+1 1]  ;; Move diagonally down-right
        set D-prob dispProbability (column-number - 1)(distD)
        if (random-float 1) <= D-prob [set tau-1-1 1]  ;; Move diagonally down-left

        set proportion (proportion + immigration-rate * (tau-0-1 * [proportion-temp] of turtle (who
             - 1) + tau-0+1 * [proportion-temp] of turtle (who + 1) +
          tau-1-0 * [proportion-temp] of turtle (who + column-number) + tau-1-1 * [proportion-temp]
              of turtle (who + column-number - 1) +
          tau-1+1 * [proportion-temp] of turtle (who + column-number + 1)))
      ]
    ]
    ]
        ;; If at the top-right corner
      [ifelse who >= (column-number * (row-number - 1))[
      ifelse who = (column-number * (row-number - 1))[
        set H-prob dispProbability (who + 1)(distH)
        if (random-float 1) <= H-prob [set tau-0+1 1]

        set D-prob dispProbability (who - column-number + 1)(distD)
        if (random-float 1) <= D-prob [set tau+1+1 1]

        set proportion (proportion + immigration-rate * (tau-0+1 * [proportion-temp] of turtle (who
             + 1) + tau+1-0 * [proportion-temp] of turtle (who - column-number) +
          tau+1+1 * [proportion-temp] of turtle (who - column-number + 1)))
        ][ifelse who = (column-number * row-number - 1)[
          set H-prob dispProbability (who - 1)(distH)
          if (random-float 1) <= H-prob [set tau-0-1 1]

          set D-prob dispProbability (who - column-number - 1)(distD)
          if (random-float 1) <= D-prob [set tau+1-1 1]
```

```
          set proportion (proportion + immigration-rate * (tau-0-1 * [proportion-temp] of turtle (
              who - 1) + tau+1-0 * [proportion-temp] of turtle (who - column-number) +
            tau+1-1 * [proportion-temp] of turtle (who - column-number - 1)))
        ][
          set H-prob dispProbability (who - 1)(distH)
          if (random-float 1) <= H-prob [set tau-0-1 1]
          set H-prob dispProbability (who + 1)(distH)
          if (random-float 1) <= H-prob [set tau-0+1 1]

          set D-prob dispProbability (who - column-number + 1)(distD)
          if (random-float 1) <= D-prob [set tau+1+1 1]
          set D-prob dispProbability (who - column-number - 1)(distD)
          if (random-float 1) <= D-prob [set tau+1-1 1]

          set proportion (proportion + immigration-rate * (tau-0-1 * [proportion-temp] of turtle (
              who - 1) + tau-0+1 * [proportion-temp] of turtle (who + 1) +
            tau+1-0 * [proportion-temp] of turtle (who - column-number) + tau+1+1 * [proportion-
                temp] of turtle (who - column-number + 1) +
            tau+1-1 * [proportion-temp] of turtle (who - column-number - 1)))
        ]
      ]
      ][ifelse (who mod column-number) = 0[
        set H-prob dispProbability (who + 1)(distH)
        if (random-float 1) <= H-prob [set tau-0+1 1]


        set D-prob dispProbability (who - column-number + 1)(distD)
        if (random-float 1) <= D-prob [set tau+1+1 1]
        set D-prob dispProbability (who + column-number - 1)(distD)
        if (random-float 1) <= D-prob [set tau-1+1 1]

        set proportion (proportion + immigration-rate * (tau-0+1 * [proportion-temp] of turtle (who
             + 1) + tau-1-0 * [proportion-temp] of turtle (who + column-number) +
          tau+1-0 * [proportion-temp] of turtle (who - column-number) + tau+1+1 * [proportion-temp]
               of turtle (who - column-number + 1) +
          tau-1+1 * [proportion-temp] of turtle (who + column-number + 1)))
        ][ifelse (who mod column-number) = (column-number - 1)[
          set H-prob dispProbability (who - 1)(distH)
          if (random-float 1) <= H-prob [set tau-0-1 1]

          set D-prob dispProbability (who - column-number - 1)(distD)
          if (random-float 1) <= D-prob [set tau+1-1 1]
          set D-prob dispProbability (who + column-number - 1)(distD)
          if (random-float 1) <= D-prob [set tau-1-1 1]

          set proportion (proportion + immigration-rate * (
          tau-0-1 * [proportion-temp] of turtle (who - 1)  ;; Rust immigration from the left
        + tau-1-0 * [proportion-temp] of turtle (who + column-number)  ;; Rust immigration from
            below
        + tau+1-0 * [proportion-temp] of turtle (who - column-number)  ;; Rust immigration from
            above
        + tau+1-1 * [proportion-temp] of turtle (who - column-number - 1)  ;; Rust immigration from
             top-left diagonal
        + tau-1-1 * [proportion-temp] of turtle (who + column-number - 1)))  ;; Rust immigration
            from bottom-left diagonal
```

```
          ][
            set H-prob dispProbability (who + 1)(distH)  ;; Compute probability of rightward movement
            if (random-float 1) <= H-prob [set tau-0+1 1]  ;; Move right

            set H-prob dispProbability (who - 1)(distH)  ;; Compute probability of leftward movement
            if (random-float 1) <= H-prob [set tau-0-1 1]  ;; Move left





            set D-prob dispProbability (who - column-number - 1)(distD)  ;; Compute probability of
                diagonal up-left movement
            if (random-float 1) <= D-prob [set tau+1-1 1]  ;; Move diagonally up-left

            set D-prob dispProbability (who - column-number + 1)(distD)  ;; Compute probability of
                diagonal up-right movement
            if (random-float 1) <= D-prob [set tau+1+1 1]  ;; Move diagonally up-right

            set D-prob dispProbability (who + column-number - 1)(distD)  ;; Compute probability of
                diagonal down-left movement
            if (random-float 1) <= D-prob [set tau-1-1 1]  ;; Move diagonally down-left

            set D-prob dispProbability (who + column-number + 1)(distD)  ;; Compute probability of
                diagonal down-right movement
            if (random-float 1) <= D-prob [set tau-1+1 1]  ;; Move diagonally down-right

            set proportion (proportion + immigration-rate * (
              tau-0-1 * [proportion-temp] of turtle (who - 1)  ;; Rust immigration from the left
            + tau-0+1 * [proportion-temp] of turtle (who + 1)  ;; Rust immigration from the right
            + tau-1-0 * [proportion-temp] of turtle (who + column-number)  ;; Rust immigration from
                below
            + tau+1-0 * [proportion-temp] of turtle (who - column-number)  ;; Rust immigration from
                above
            + tau+1-1 * [proportion-temp] of turtle (who - column-number - 1)  ;; Rust immigration
                from top-left diagonal
            + tau+1+1 * [proportion-temp] of turtle (who - column-number + 1)  ;; Rust immigration
                from top-right diagonal
            + tau-1-1 * [proportion-temp] of turtle (who + column-number - 1)  ;; Rust immigration
                from bottom-left diagonal
            + tau-1+1 * [proportion-temp] of turtle (who + column-number + 1)))  ;; Rust immigration
                from bottom-right diagonal
          ]
        ]
      ]
    ]
  ]
end

to-report dispProbability [num d]
  ;; Calculates the dispersal probability of bacteria using a Gaussian function
  report prop-prob * ([proportion-temp] of turtle num) * exp(-1 * d ^ 2 /(2 * sigma ^ 2)) / (2 * pi *
       sigma ^ 2)
end

to grow-bacteria
```

```
  ask turtles[
    ;; Update the bacteria size based on logistic growth model

    set bacteria-size (bacteria-size + bacteria-size * (growth-rate-bacteria * (1 - bacteria-size /
        carrying-cap-bacteria) - death-rate))

  ]
  ask turtles[control-strategy]  ;; Applies a control strategy to each turtle
end

to control-strategy
  ifelse global-control ;; Check if global control is enabled
  [
    ;; Global control: Apply a uniform increase to all bacteria
    set bacteria-size (bacteria-size + main-irrigation)
  ]
  [
    ;; Local control: Adjust bacteria size based on the position of the turtle
    ifelse who < column-number [
      ;; Top row turtles
      ifelse who = 0 [
        ;; Top-left corner
        set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation * ([
            proportion-temp] of turtle column-number + [proportion-temp] of turtle 1))
      ] [
        ifelse who = (column-number - 1) [
          ;; Top-right corner
          set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation * ([
              proportion-temp] of turtle (2 * column-number - 1) + [proportion-temp] of turtle (
              column-number - 2)))
        ] [
          ;; Other top-row turtles
          set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation * ([
              proportion-temp] of turtle (who - 1) + [proportion-temp] of turtle (who + 1) + [
              proportion-temp] of turtle (who + column-number)))
        ]
      ]
    ] [
      ifelse who >= (column-number * (row-number - 1)) [
        ;; Bottom row turtles
        ifelse who = (column-number * (row-number - 1)) [
          ;; Bottom-left corner
          set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation * ([
              proportion-temp] of turtle (who + 1) + [proportion-temp] of turtle (who - column-
              number)))
        ] [
          ifelse who = (column-number * row-number - 1) [
            ;; Bottom-right corner
            set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation *
                ([proportion-temp] of turtle (who - 1) + [proportion-temp] of turtle (who - column-
                number)))
          ] [
            ;; Other bottom-row turtles
            set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation *
                ([proportion-temp] of turtle (who - 1) + [proportion-temp] of turtle (who + 1) + [
                proportion-temp] of turtle (who - column-number)))
```

```
          ]
        ]
      ] [
        ifelse (who mod column-number) = 0 [
          ;; Leftmost column turtles
          set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation * ([
              proportion-temp] of turtle (who + 1) + [proportion-temp] of turtle (who + column-
              number) + [proportion-temp] of turtle (who - column-number)))
        ] [
          ifelse (who mod column-number) = (column-number - 1) [
            ;; Rightmost column turtles
            set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation *
                ([proportion-temp] of turtle (who - 1) + [proportion-temp] of turtle (who + column-
                number) + [proportion-temp] of turtle (who - column-number)))
          ] [
            ;; All other turtles
            set bacteria-size (bacteria-size + main-irrigation * proportion + secondary-irrigation *
                ([proportion-temp] of turtle (who - 1) + [proportion-temp] of turtle (who + 1) + [
                proportion-temp] of turtle (who + column-number) + [proportion-temp] of turtle (who
                - column-number)))
          ]
        ]
      ]
    ]
  ]
end
```

Listing .1 Source code of the simulation model in NetLogo