BIO247 - Final Paper

Elijah Muzzi

Seaweed Competition Simulator

Introduction

Hawaii has a very unique reef ecosystem with many endemic wildlife found only in the Hawaiian Islands. A lot of these species evolved to live together but left some environmental niches unfilled. Due to many reasons like being brought over by boats or on shipping containers or even being released by aquarium enthusiast, invasives species of seaweed have abused the absense of certain niches and have taken over spaces where endemic seaweed used to thrive. This phenomena has been taken to the exterme in places like Kaneohe Bay where invasive seaweeds have pretty much taken over the entire area. Removal of the seaweed used to be a tedious process, “On average, it would take a diver two strenuous hours to remove one square meter (roughly 10.5 square feet) of the exotic red algae carpeting coral reefs…”. The solution to this came in the form of the Super Sucker, a 40 horse power underwater vaccum that divers would use to suck up the invasive seaweed. This made a noticable dent in the invasive seaweed population but there was hopes for a more natural control, reintroducing preditors of the invasive seaweed. Collector Urchins are native to Hawaii and we recenly figured out how to breed them in captivity.

Based on this real life scenerio, I asked the question: “How do these control methods impact the competition of two seaweeds competing for space?” and additionally “What is the optimal method for keeping the invasive population in check?”

To answer these questions I designed an agent based discrete time simulation of two seaweed agents growing in a limited space.

Methods

My model for seaweed growth is based on the BESeM, Bio Economic Seaweed Model, which describe a set of equations that was created based on experimental data collection. I was particulalry interested in an equation for weight as a function of time since planted.

$$w_{f,g}\left(t\right) = \frac{w_{f,\max}}{1+\left(\frac{w_{f,\max}-w_{f,0}}{w_{f,0}}\right)*e^{-rgrmax*t}}$$

Figure 1

To take advantage of this equation, I needed to create Seaweed agents that can keep track of the information needed for this equation.

$w_{f,g}$ = weight in Kg after t days since planted
$w_{f,max}$ = maximum weight of the species of seaweed
$w_{f,0}$ = initial weight of seaweed (weight at t=0)
$rgrmax$ = growth rate of the seaweed species

To do this, I created a seaweed class that takes all of these as input to a constructor

```
class seaweed:


    #constructor
    def __init__(self, name,initialSize, growthRate, maxSize):
        self.name = name
        self.currentSize = initialSize
        self.initialSize = initialSize
        self.growthRate = growthRate
        self.maxSize = maxSize
```

Figure 2

After this, I created an update function that uses the equation in Figure 1

```
# update is a function of days since planting in relation to biomass
def update(self,t,x):
    newBiomass = (self.maxSize)/((1)+(((self.maxSize - 
self.initialSize)/self.initialSize)*math.exp(self.growthRate*-1*t)))
    if newBiomass < self.maxSize:
        if newBiomass > x:
            self.currentSize = x
        else:
            self.currentSize = newBiomass
    else:
        self.currentSize = self.maxSize
```

Figure 3

The parameters this function takes is self (which is passed when it is called) t which is either the days or lastgrown depending on the state of the simulation, and x if the amount of space left for self to grow relative to the amount of space that the other species of seaweed is taking up. There are some additional logical checks in Figure 3, the first is to check if the newBiomass of the seaweed is less than the maximum size of that species. If the new biomass is larger than the max size, it will set the current size to the maximum size. If the first condition is true, it will then go on to check if the new biomass is greater than the size of the space to grow into (x).

Now that we have the Seaweed agent set up, we can get into the simulation logic:

```
#------------------------
# parameters
# t is the maximum number of timesteps you want to simulate in days
# mode is the types of population controls you want to impose on N2 (species
2 of seaweed)
# size is the maximum amount of space the two seaweeds have to grow into (the
combined size of both has to be less than size)
# agents is the list of seaweed agents that will be present in the simulation
# freq is the frequency the manual removal method will happen
# percent is the percent of the total biomass of seaweed that is removed
every time manual removal occurs
#---------------------------

def sim(t,mode,size,agents,freq,percent):
    days = 0
    lastState = True    # Keeps track of the last state to know when to
update lastGrown
    x = size            # Total size that both seaweed agents have to share
    xaxis = []  #sets up an array to capture timesteps as we move through the
simulation
    N1Trace = []   # sets up an array to capture the values N1 biomass as we
move through time
    N2Trace = []    # sets up an array to capture the values N2 biomass as we
move through time
    N1 = agents[0]
    N2 = agents[1]
    adjValue = 0    # This will either hold the value of days or lastgrown
depending on the sequence of states
    lastgrown = 0
    firstTime = True   # First time lastgrown is updated this is set to false
    notAdjusted = True # If the growth rates have not yet been adjusted for
the collector urchins, it will adjust them then set this to false
```

Figure 4

The code in figure 4 is the part of the sim function that creates all the data structures we will use later in the simulation as well as instantiates the variable.

```
 while days < t:
     leftToGrow = ((N1.currentSize + N2.currentSize) < x) # checks at each
timestep to see if there is room left to grow
     #print(leftToGrow,days,lastgrown)
     if lastState == True and leftToGrow == False:   # Updates lastgrown to
be the day that the plant last was able to grow
         if firstTime == True:   # The first time this happens, lastgrown
will be equal to days-1
             #print("First Time True to False Change")
             lastgrown = (days-1)
             #print(lastgrown)
         else:
             #print("Incrimenting lastgrown...")
             lastgrown += 1
             #print(lastgrown)
     if lastState == True and leftToGrow == True:    # Updates lastgrown when
it grew both this state and the previous state
         #print("Incrimenting lastgrown...")
         lastgrown += 1

     if lastState == False and leftToGrow == True:   # If we could not grow
for a while and then change back to true
         #print((N1.currentSize + N2.currentSize))
         if firstTime == True:
             #print("First Time False to True Change")
             firstTime = False
             lastgrown += 1
             #print(lastgrown)
         adjValue = lastgrown    # While it hasn't grown for a while, the
next time it can grow adjValue should be equal to lastgrown

     if lastState == True and leftToGrow == True:
         #print((N1.currentSize + N2.currentSize))
         if firstTime == True:           # While it hasn't not grown,
adjValue should be equal to days
             adjValue = days

NameError: name 'days' is not defined
```

Figure 5

The code in figure 5 is the while loop of our simualtion function. Days was instantiated as 0 and get incremented by one every interation of the loop. Once days is equal to the desired time to run the simulation the loop breaks. While we are inside of the loop, We set left to grow to a boolean value to check if there is any space left to grow for either species of seaweed. Then we have a series of if else statement that keep track of the current and last state of the simulation and react appropriately. The first statement checks to see if the seaweed grew on the last state and can not grow on this state. In this case, we want to keep track of the last day that the seaweed grew since the equation is a function of days since planting and we get weird behavior if it jumps from the last time it grew to the next time step it can grow. The last grown variable was our answer to this problem. If the seaweed grew both this timestep and last timestep, we increment lastgrown by one. The first time it goes from not being able to grow to being able to grow, we set the adjValue variable to hold lastgrown instead of days. It is best to use days as the value for adjValue until we get to case where using days is incorrect and we need to start keeping track of and using lastgrown as the adjValue.

```
 if (mode == 'Manual Removal' or mode == 'Removal + Urchin') and days%freq == 
0:
     N2.remove(percent)
 if mode == 'Collector Urchin':
     if notAdjusted:
         N1.adjRGR()
         N2.adjRGR()
         notAdjusted = False
 if mode == 'Removal + Urchin':
     if notAdjusted:
         N1.adjRGR()
         N2.adjRGR()
         notAdjusted = False

NameError: name 'mode' is not defined
```

Figure 6

The code in figure 6 is to apply the right population controls on species 2. It checks the value of the mode paraemter passed into the sim function when we start and makes sure that it is implimenting logic that reflects the effects of the control on the system. If we select manual removal or removal + urchin as the mode, we will remove the specified percent at the specified frequency. If the mode is Collector Urchin, we call the adjRGR() function which adjust the growth rate to reflect preditation by a suffciently large population of collector urchins. I am using data collected from a study that says anually Collector Urchins will eat 24% of the total biomass of the seaweed in the area. Not only am I assuming that the population is sufficently large but I am also assuming that the annual rate applies over any period of time. This assumption is slightly incorrect because Collector Urchins feed a lot more between November and January and less throughout the year. Since my simulation does not account for seasonal differences, I think this is not a harmful assumption to make.

```
if N1.currentSize < N1.maxSize:
    if leftToGrow:
        N1x = N2.relativeX(x)
        N1.update(adjValue,N1x)

if N2.currentSize < N2.maxSize:
    if leftToGrow:
        N2x = N1.relativeX(x)
        N2.update(adjValue,N2x)
```

```
NameError: name 'N1' is not defined
```

Figure 7

The code in figure 7 is the part of the sim function that checks how much space a seaweed agent has to grow based on the relative amount of space left. This and some of the logic in the update function enforces conservation of seaweed agents and makes it so we can't have anything growing outside of the space confines that we defined in the size parameters. (Size can either be thought of as a numberline where Species 1 starts at 0 and Species 2 starts at size and they grow towards each other or as a Total Biomass Cap)

```
    N1Trace.append(N1.currentSize)
    N2Trace.append(N2.currentSize)
    xaxis.append(days)
    lastState = leftToGrow
    days += 1

plot(xaxis,N1Trace,N2Trace,N1,N2)

NameError: name 'N1Trace' is not defined
```

Figure 8

In figure 8 we have the end of the simulation logic where we appended the data to the relevant data structre and keep track of the biomass over time of each agent. After we break out of the loop we then call the plot function.

```
#------------------------
# parameters
# x is an array of values for the xaxis
# values1 and values2 are arrays for the values of the biomass of seaweeds 1
& 2 over time
# N1 and N2 are the seaweed agents so that the legend can use the getName
function
#---------------------------
def plot(x,values1, values2,N1,N2):
    plt.style.use('Solarize_Light2')  # Using a predefined style for nicer
plots
    plt.cla() # clears plot
    plt.ion() # makes interactive
    plt.plot(x, values1, label=f'{N1.getName()} Biomass')
    plt.plot(x, values2, label=f'{N2.getName()} Biomass')
    plt.legend()
    plt.title("Trace of Seaweed Biomass over Time")
    plt.xlabel('Time (Days)')
    plt.ylabel('Biomass (KG)')
    plt.show()
```

Figure 9

Results

In my original question, I wanted to see what control method was most beneficial for native Hawaiian species of seaweed. Through out the course of my project I had to adust my question due to lack of online data about the maximum size and relative growth rates for these endemic species. In further work I would like to work with the University of Hawaii Manoa Marine Biology Department or some other organization and see if any of this data is available or conduct my own study in my free time. My adjusted research question is to see how different population control methods impact the seaweed populations. I will look at low vs high frequency removal, moderate vs aggressive percentage removal, the effect of Collector Urchins and how these two methods can work together.

Validation

After the creation of my model, I will compare my output to expiremental data from the BESeM expirement

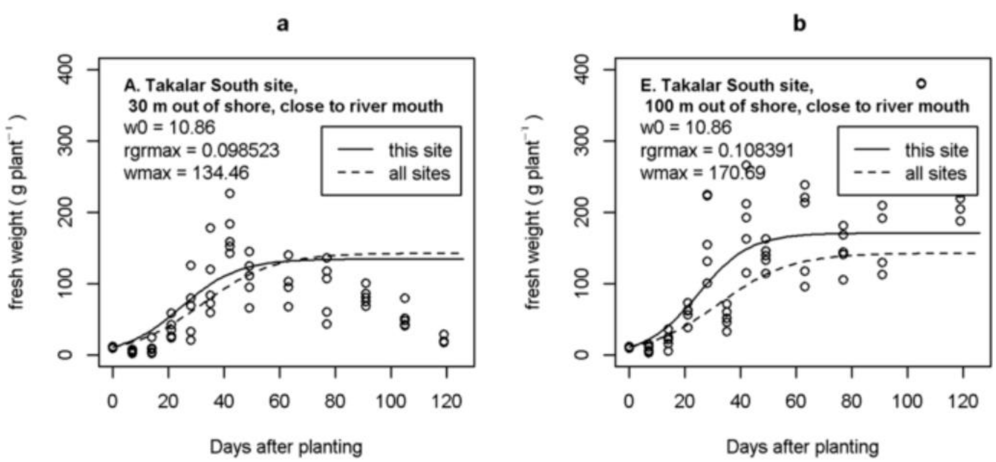

Figure 10

Using seaweed agents with the same max weight, initial weight and growth rate I was able to recreate these curves on the same graph with my Seaweed competition simulator

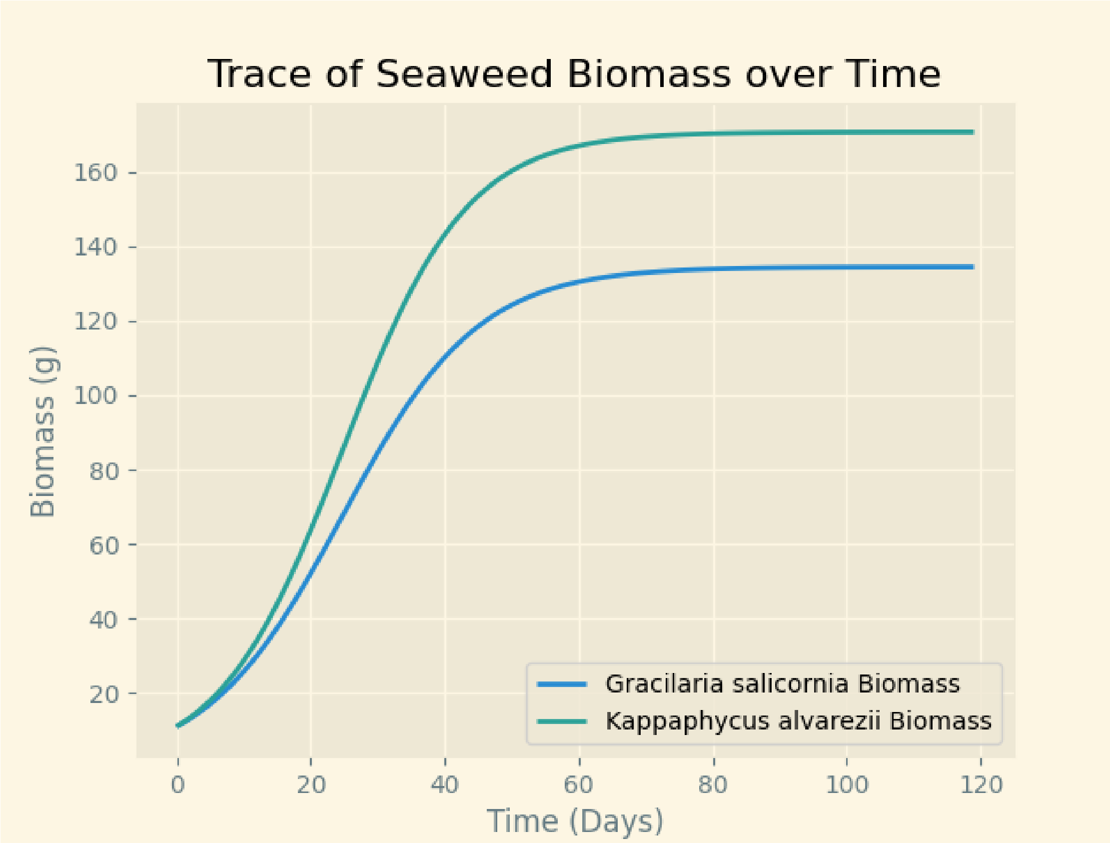

Figure 11

In Figure 11, the lines from Figure 10 a and b are put on the same plot and seem to confirm that the model correctly captures expiremental data of seaweed growth. The BESeM model also havisting of the seaweed and goes into how profitable growing certain species of seaweed in certain conditions will be. I adapted this model to instead of 'harvesting' seaweed we are 'removing' without needing to keep track of the other more econimic side of the model. In the next section, we will look at the effects of different population control methods on these two species of seaweed.

Manual Removal

The manual removal population control mode is made to simulate the 'super sucker' that is being used for seaweed population control in Kaneohe Bay, Oahu. I allowed for customization of the frequency of and ammount remove so that the people involed in deciding when to use the super sucker and how much they want to remove while they are using it. This will hopfully give insight to the optimal way to use this tool.

Limited vs Unlimited Space

When doing my research, I found that the population control measures have the greatest impact on systems with limited space and if the space available is so much that it can be considered 'limitless', it is really hard to notice the impact. Going forward we will hold the assumption that space is limited and the maximum size of both the seaweed speices is more than the availble space.

Low vs High Frequency

I will be comparing the effects of removing 5% of species 2 every week vs every other month over the course of a year. The first figure will be the parameters that I input into the GUI to make the corresponding graph output.

Figure 12

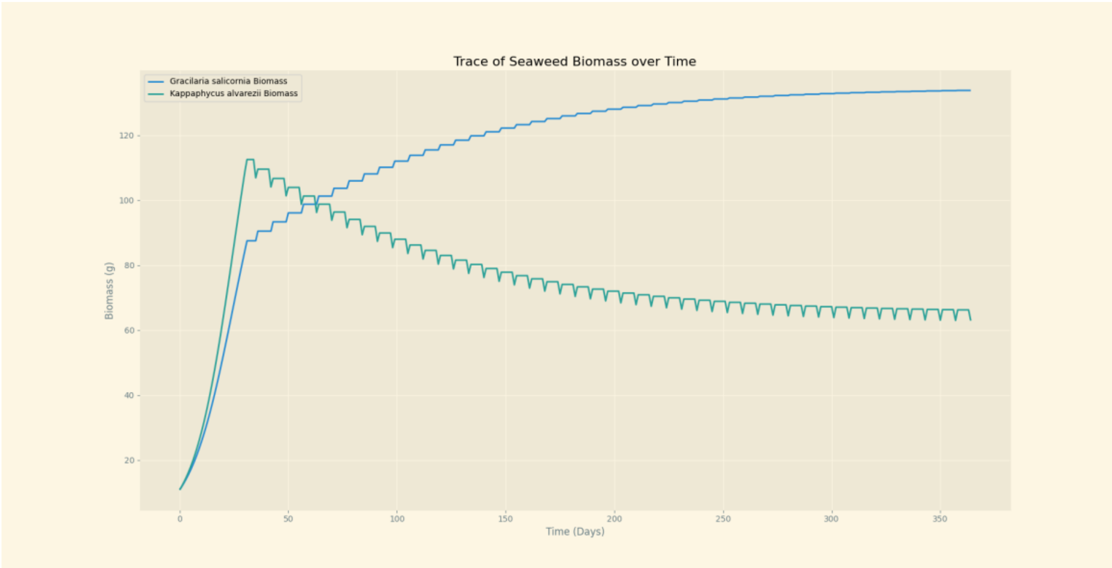

Figure 13

In Figure 13 we see the graph of manual removal of 5% of the total biomass every 7 weeks for a year. If the blue is the species we want to help support, this graph seems to indicate that this would be an effective strategy since the blue line is allowed to approach its carrying capacity while the green line starts to pleateau around 1/2 carrying capacity. This is important because if left alone like in Figure 11, the green species will grow faster and take over.

Figure 14

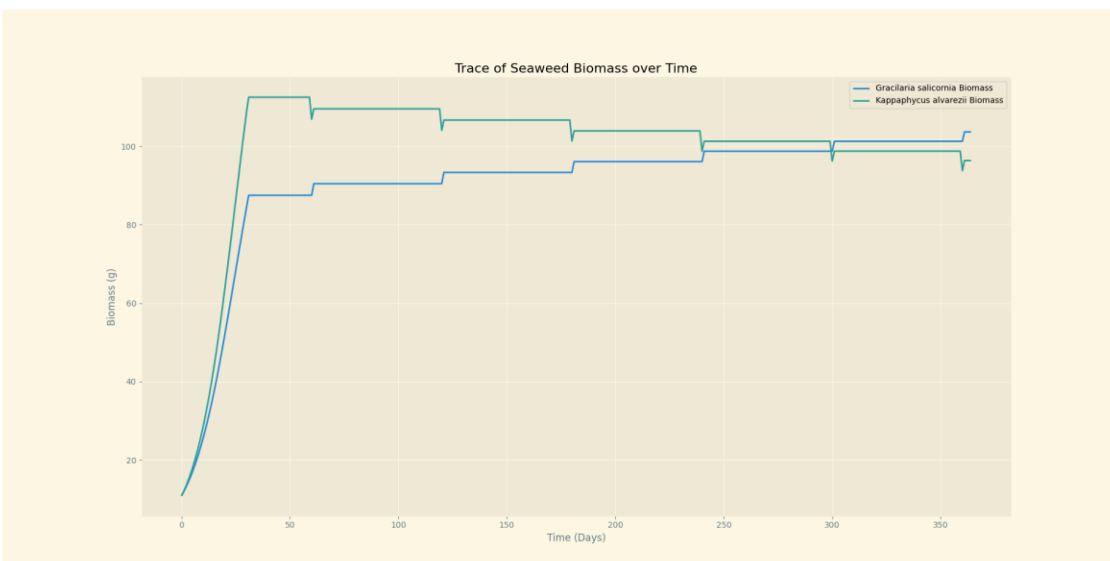

Figure 15

In Figure 15 we see that the every other month removal creates step like behavior vs the curve we see in high frequency removal. It would take about 10 months of remvoing 5% every other month until species overtakes species 2. This result leads to nearly equal populations of species 1 and 2 by the end of the year. This approach may be more viable due to real world conditions such as cost or divers available to go out and use the super sucker.

Moderate vs Aggressive Removal

If we hold the time between removals constant (every month), we can try to figure out the optimal amount to remove to get the most value out of the divers time in the wild. I will try 5%, 10% and 25% removal every month for a year and analyze the results.

Figure 16

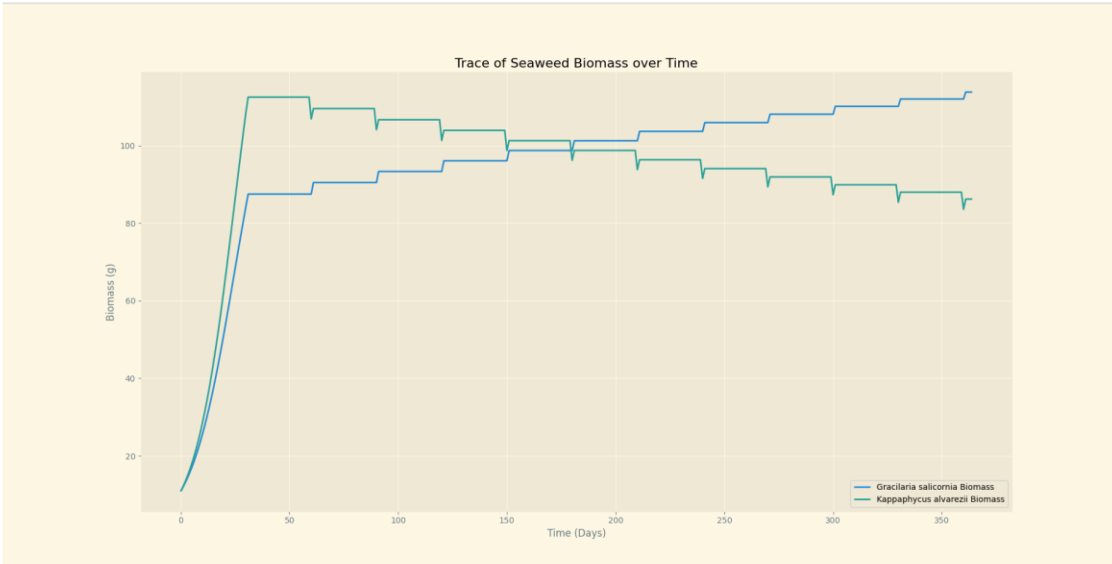

Figure 17

In Figure 17 we can see that at month 6, species 1 overtakes species 2 and it seems to continue to grow further and further apart as the year goes on.

Figure 18

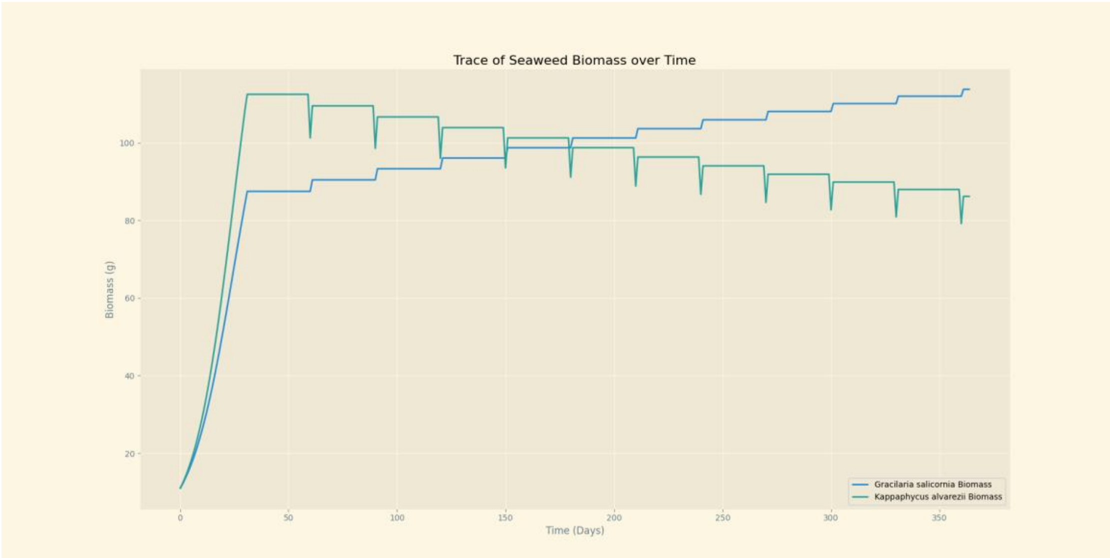

Figure 19

The most intresting thing from Figure 19 is the fact that we double the amount removed each time the super sucker used but even that increase in removal leaves us in the same situation as Figure 17. We stil see species 1 overtake species 2 at 6 months in and we end up with a similar ratio of species 1 to species 2. This graph seems to indicate that the amount at least between 5% and 10%, there is not much addition benefit to doing double the work.

Figure 20

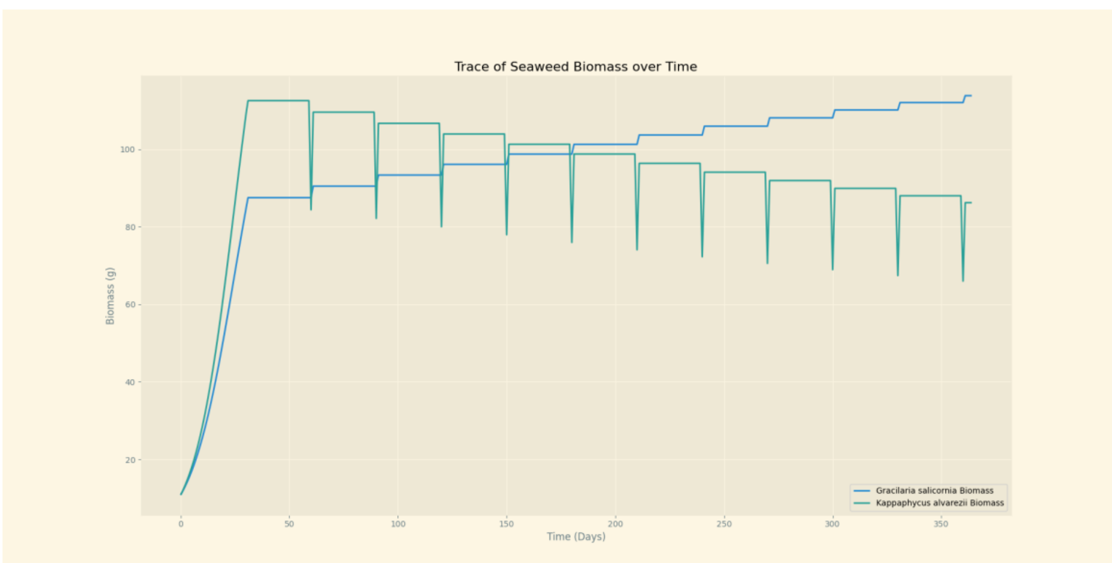

Figure 21

Figure 21 showcases that the growth of species 2 is so rapid that even a pretty aggressive removal of 25% of the biomass every month still allows for the seaweed to regrow faster than species 1 can to claim some of that space. After looking at Frequency vs Percent removal, we can see that the frequency of the removals has a greater impact on the ratio of species 1 to species over time. Due to the rapid growth of some species of invasive seaweeds, even 25% is not enough to make a difference in at which point species 1's biomass overtakes species 2 and the releative amounts of each at the 1 year mark. Anything too much greater than 25% is unrealistic with the amount of people they have to do these clean up jobs and the large amount of area they would have to cover.

Collector Urchins

Tripneustes gratilla, or Collector Urchins, are native to Hawaii and have recenlty been bred in captivty. They are natural preditors of invasive seaweed and an effort has been made to breed them and release them into Kaneohe Bay. In my research, I found that a sufficently large population of Collector Urchins will eat up to 24% of the total seaweed population. To simulate this, I adjust each species growth rate to reflect a 24% decrease vs a population with no urchins present. For this simulation, I will change the size back to 'unlimited' so that we can compare the Urchin graph with the validation graph since the only difference between the two should be the growth rate adjusted for urchins.

Figure 22

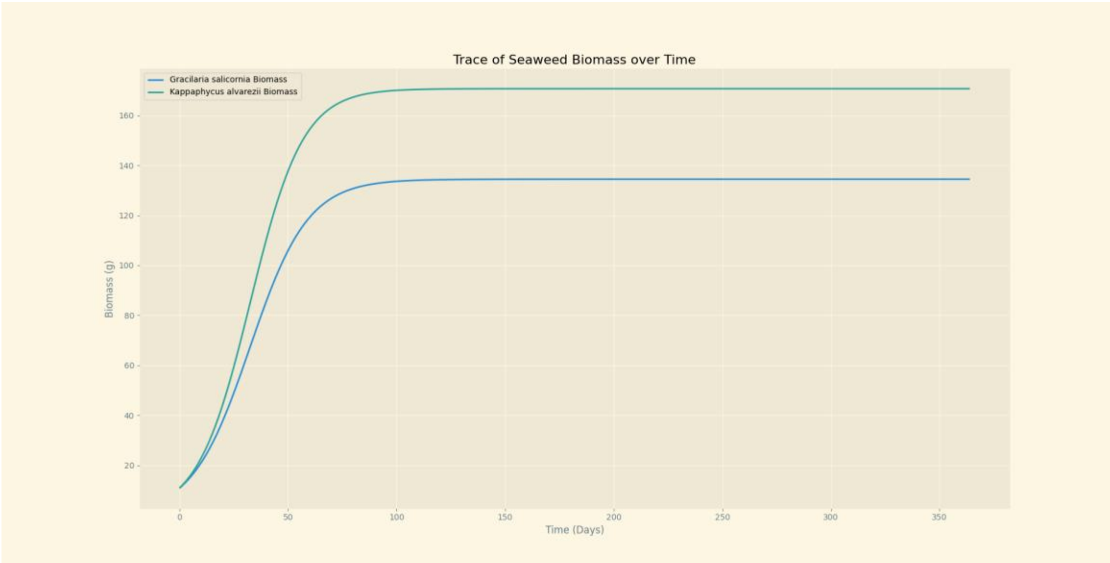

Figure 23

Figure 23 is very similar to Figure 11, the main difference being the slope is slighlty less steep and the graph levels off slightly later.

Manual Removal and Collector Urchin

If we look at a graph of both population control methods used together we can see it has an interesting effect.

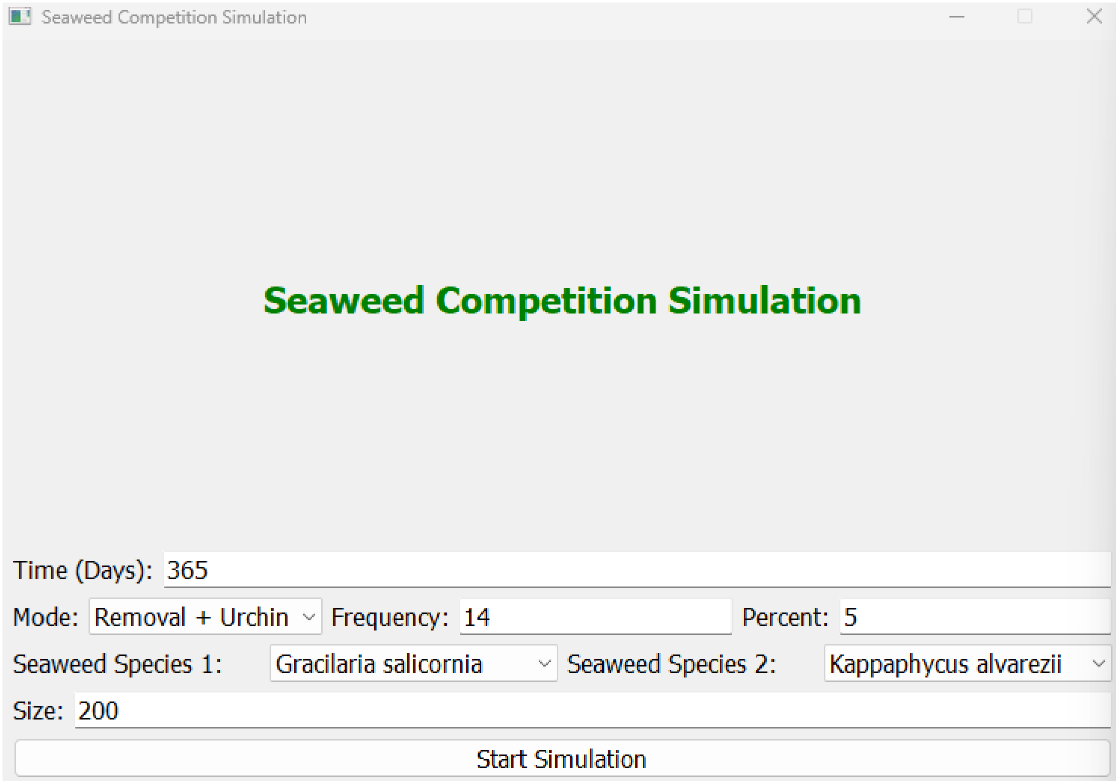

Figure 24

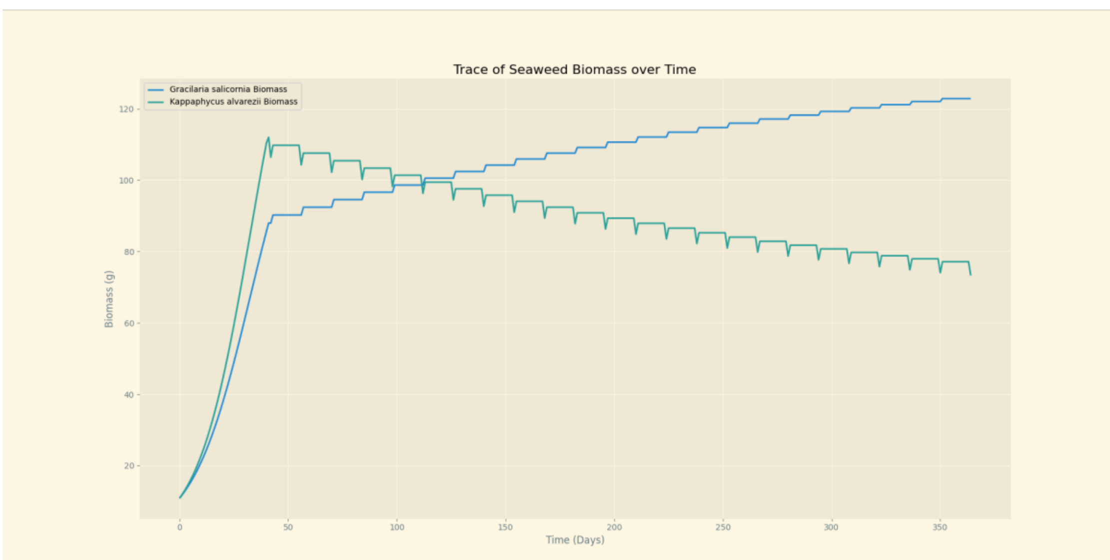

Figure 25

When using both techniques together, we can that the time between removal and recovery has increased slightly which allows species one to grow a little bit more beofre species 2 reclaims that space.

Discussion

In my exploration of my research question I found some interesting trends I did not expect. Going into my experiment I had a feeling that having a more agressive less frequent super sucker strategy would have been one of the more effective methods of population control. However, my simulation showed that a less agreesive removal of 5% of the total biomass of seaweed 2 every week actually had a more significant impact on ratio of species 1 to species 2 and how fast speicies 1 became the dominant species. Species 1 overtook speice 2 almost tree times as fast with the less agressive frequent approach. Figure 21 is a great example of why this is the case, the growth rate of species 2 is so high that is is able to retake the space way faster than speicies 1 and there is almost no change in the ratio of species 1 to species 2. In the case of only using Collector Urchins as a mean of population control, we see that Figure 23 looks very similar to Figure 11 where there is no means of population control. This initially made me think that it might not be as effective as the super sucker but there was a slight reduction in the steepness of the slope and how fast they reached the point where the lines both flattened out. The biggest benefit of this growth rate reduction comes from using the super sucker in conjuction with maintaining a sufficently large population of Collector Urchins. If we compare Figure 21 with Figure 25 we see that the valleys that occur when we manually remove some of the biomass start to get wider in Figure 25. This is becuease the reduce growth rate means that species 2 takes more time to fill up the space that the removal left behind which allows species 1 more time to grow into it than just manual removal alone. This model is a simplifactation of its real world counterpart and it does not take into account that there could be more than two speices in a space, how these interaction may change in 3d space, seasonal differences, other prediation on seaweed species any many many other factors that make up our complex ecosystems. Going forward, I would like to confirm some of these patterns I have seen through expiremental data. I can do this by having two species of seaweed in a confined space and measure their weight every day over a period of time. I would have one container as a control where there is no population control method, and one container for the various strategies that produced intresting data and see if I can recreate those patterns. Once these results are again validated with real world systems and I know the model is not losing too much through its simplifacation, I would love to scale up the model by adding more species of seaweed. Specifically, I would like to collect data and add species that are native to Hawaii so I can answer my orginal question of how to best protect our endemic sea life.

References

“15.5: Quantifying Competition Using the Lotka-Volterra Model.” *Biology LibreTexts*, Libretexts, 25 Mar. 2024, bio.libretexts.org/Courses/Gettysburg_College/01%3A_Ecology_for_All/15%3A_Competition/15.05%3A_Quantifying_Competition_Using_the_Lotka-Volterra_Model.

“Sea Urchin Hatchery.” *Sea Urchin Hatchery*, dlnr.hawaii.gov/ais/invasivealgae/urchn-hatchery/. Accessed 8 May 2024.

“Super Sucker.” *Super Sucker*, dlnr.hawaii.gov/ais/invasivealgae/supersucker/. Accessed 8 May 2024.

“Tripneustes Gratilla.” *Wikipedia*, Wikimedia Foundation, 28 Jan. 2022, en.wikipedia.org/wiki/Tripneustes_gratilla.

van Oort, P A J, et al. “The Bio Economic Seaweed Model (BESEM) for Modeling Tropical Seaweed Cultivation - Experimentation and Modelling.” *Journal of Applied Phycology*, U.S. National Library of Medicine, 2022, www.ncbi.nlm.nih.gov/pmc/articles/PMC9390115/.

Xiao, Xi, et al. “Dataset on the Specific Growth Rate of Seaweed Growing in Experimental and Aquaculture Conditions at Various Biomass Densities, Irradiation and Nutrient Concentrations.” *Xiao, X et al. (2019): Dataset on the Specific Growth Rate of Seaweed Growing in Experimental and Aquaculture Conditions at Various Biomass Densities, Irradiation and Nutrient Concentrations*, PANGAEA, 10 Oct. 2019, doi.pangaea.de/10.1594/PANGAEA.907407.