Sharai Wilson
4/23/19

Introduction

In the year 2014, two papers on social spiders were published that, while tackling questions on separate scales, both shared an author and underlying theory.

The first, "*Animal personality aligns task specialization and task proficiency in a spider society"* by Wright, Holbrook, and Pruitt (2014), categorizes individual *Anelosimus studiosus* specimen as being either aggressive or docile based on how close they'll sit to another specimen in a box and tests them to see how well or how likely they are to perform one of four tasks: colony defense, prey capture, web repair, and parental care. Their results showed that aggressive spiders are better at and will do more often the former three tasks, while docile spiders are specialized in parental care.

The second, *"Site-specific group selection drives locally adapted group compositions"* by Pruitt and Goodnight (2014), studied the personality composition (ie. aggressive/docile) of both native and experimental *Anelosimus studiosus* colonies at several different natural sites with either low or high levels of resource abundance. They found that the composition of personality in a given colony depends on both the size of the population and the resource levels of the colony site. Small colonies in high-resource sites and large colonies in low-resource sites have more docile spiders, white large colonies in high-resource sites and small colonies in low-resource sites have more aggressive spiders.

An unspoken assumption hides in the connection one is inclined to draw between these two papers: spider personality is relevant to colony adaptation and survival because of the correlation (if not causation) it has with what the spiders are specialized in, which in aggregate determines how well suited the colony is towards fulfilling its needs. It then begs exploring whether there is a causal link or if there may be some other third factor driving the correlation, and that is exactly what this model is here to try to find.

To see that through, the model is built and designed to (attempt to) do the following:

a. Have individual spiders perform, based on personality (aggressive vs. docile), tasks at frequency and proficiency similar to the findings of Wright, Holbrook, and Pruitt (2014)

b. Survive, thrive, die out, and adapt based on initial starting conditions and aggressive:docile spider ratios, similar with the results for experimental native spider mixtures in Pruitt and Goodnight (2014).

As a side thing, I'll also be exploring with different ways of implementing the aggression:docile personality spectrum to see if changing how spiders are distributed along it have a noticeable effect on the results, all else being equal (including the number of spiders on each side of the middle line).

Methods

Non-technical Description

The spiders of our simulated colony start out by sitting in a tight circle around the center of a toroidal world made out of 51x51 blank patches of space. Typically in the first several moments of their artificial lives, a good proportion of our spiders will get to work on building the web for the colony, creating one 3x3 square of webbing at a time. Once the beginning of a web is established, however, these spiders will begin to take on different tasks.

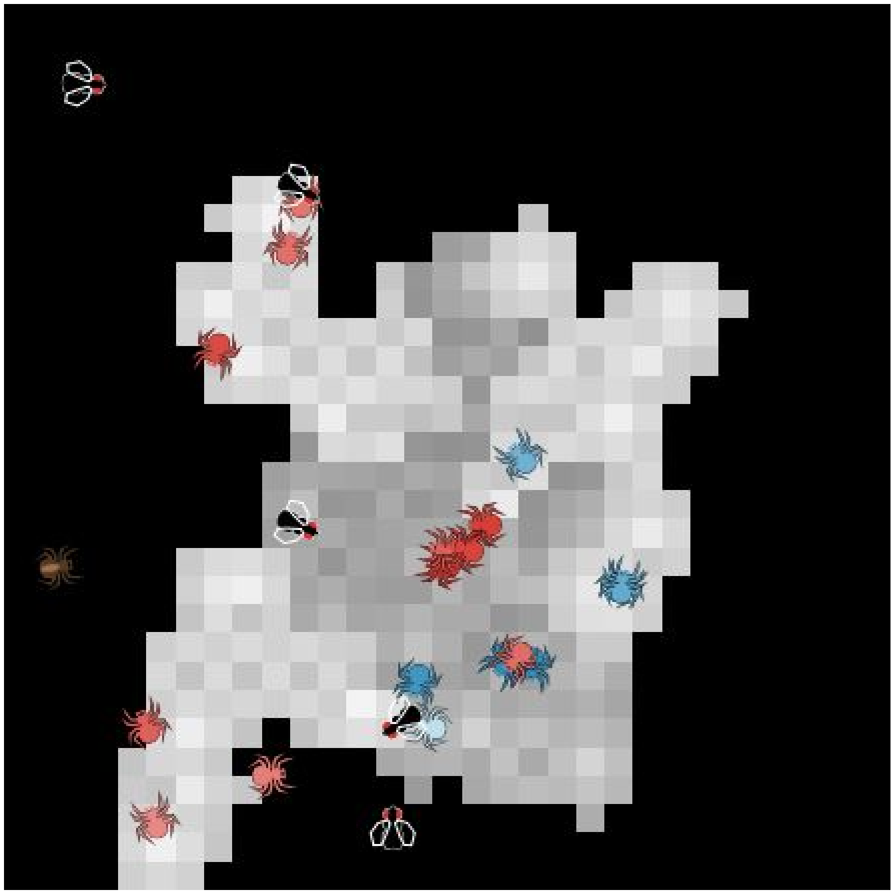
The aggressive, red spiders are highly active and like to spend their time repairing the web, catching trapped flies, and chasing away inquiline spiders. By contrast, the docile blue spiders don't fret so much, preferring to calmly spend their time raising baby spiders (which are not simulated directly, but assumed to exist) to adulthood. You may also occasionally see very light blue, light red, or even white spiders - these spiders are neither all that aggressive nor that docile, although they may lean more towards one way or the other. However, any spider may take on any task, even if they may not be as good at it as a spider more accustomed to the task!

The flies that inhabit this world are, in comparison to our social spiders, simple and humble creatures. They may drop in to visit for a time, buzzing around randomly unless (or until) they get unlucky enough to get stuck on a webbed patch! If lady luck shines on them they may escape (especially if the web is old and dingy) and fly away to live another day, but more often than not they get gobbled up by a hungry spider. If said hungry spider is one of our colony spiders, that spider will eat until they're full and share the rest of the food with the colony; this is how the colony can support spiders who rarely or never capture prey.

Speaking of hungry spiders, not all spiders are as socially inclined as our favored subjects. Sometimes opportunistic inquiline species (represented in the model as brown spiders with a light stripe) can appear on the outskirts of the colony, usually looking for an easy meal or several. They will gobble up the flies the web catches unless they are driven off or gobbled up by the colony spiders.

Unless the colony spiders dies off from starvation or old age (the latter of which can be toggled on or off), the simulation can go on indefinitely, surviving for as many generations as the limits of your patience and hardware can go.

The theory (and main hypothesis) being tested in this model is that it is the influence of personality on a spider's task preference and performance that, in aggregate, lead to the survival of and adaptation towards certain ratios of aggressive:docile spiders in spider colonies. Additionally, in Wright, Holbrook, & Pruitt (2014) and Pruitt & Goodnight (2014), personality was categorized as a binary, but might it not be that spider personality is more of spectrum like it is for us? To try to answer this question, this model is also designed with the idea in mind to explore the consequences of conceptualizing personality as a binary or a continuum in this model, all else being equal.



A zoomed-in screenshot of the model, showing several social spiders (in various shades of red and blue), some flies, an inquiline spider (to the left), and the colony's web.

Technical Description

AGENTS

Let's start with some description of the agents of the model, of which there are three: spiders, prey, and inquilines. All agents in this model are finite-state machines, but of them only the spiders have more than two or three states.

Spiders are the principle agents of the model, represented as red, blue, and white spiders in the interface. They're colored based on their *aggression* score, a variable between 0 and 100; blue for low *aggression*, red for high *aggression*, and white or near-white for middling *aggression*. The *aggression* score of a spider determines how likely they are to pick certain tasks (i.e. switch to specific states) and how well they'll perform at those tasks. Spiders also remember their *gen*eration number, *energy*, current *target* and *plan*, and their *age*.

Prey (represented in the interface as flies) are agents that have a chance (*prey-abundance*, defined in the interface) to spawn anywhere in the environment each tick. They wander around the map randomly until they either "fly away" (which has a chance of happening once their age goes past a certain threshold) or they get stuck on a sufficiently webbed patch, at which point the relevant variable (*stuck?*) gets set to true.

Prey that are stuck can wiggle around but cannot move forward, and are available to be selected as a *target* of an inquiline or spider so long as they remain stuck. Stuck prey have a chance to become free again each tick, so long as the *webbing* of the patch their on gets low enough. All prey have an *energy* variable that doesn't change over time and is transferred to the spider or inquiline that *eat*s it -- prey energy is calculated by parameter *base-prey-juiciness* * their *size* (which at this point in time is always 2).

Inquilines (represented in the interface as dark brown spiders with a light brown stripe) are antagonistic agents that have a chance (*inquiline-abundance*, defined in the interface) of spawning at the *edges* of the environment each tick. If they are not currently being chased off (i.e. *run-away?* Is false) and there is no stuck prey, they will move towards patches with higher *webbing* values. Of course, if there is a stuck prey somewhere in the environment, the inquiline will move towards it and, if it reaches it in time, *eat* it (unlike spiders, inquilines always succeed in eating prey). If the inquiline's been set into run-away mode, such as by a scary defensive spider getting too close, they will move towards the closest edge patch, where they will despawn.

Like spiders, inquilines have an *energy* variable, which increases when they *eat* prey, decrements each tick, kills them if it hits zero, and can be bequeathed to the spider that *eat*s them. Inquilines also keep track of the prey they are *target*ing (if any) and the list of spiders that are *target*ing them (called *hunters*).

ENVIRONMENT

The environment of the simulation is a toroidal world of 51x51 patches, each of which has its own *webbing* variable. *webbing* can range between 0 and 100, incrementing when a spider on or next to the patch runs the relevant section of the build-web procedure, and decrementing each tick by *web-fray-rate*%. Patches are colored in grey-scale based on their *webbing* value, with black for very low values, white for very high values, and appropriate shades of grey in-between. By default, every patch has zero *webbing* to start.

The patches that are on the edge of the world view in the interface are assigned to a patchset *edges*, which can be called when appropriate.

PARAMETERS

In the interface, there is:

- *aggression-distribution*, which can have the values "binary", "bimodal", "normal", or "flat". This determines how the aggression values of the starting population are distributed, and can have an effect on how future generations of spiders inherit aggression (in the case of binary mode)
- *initial-population*, an integer value that determines how many spiders are created at setup. The slider goes from 1 to 200, and is left at 20 by default.
- *init-aggression-mean*, the mean used to initially generate the aggression of spiders in the starting population. Can range between 0 and 100, and is left at 50 by default. The aggression first generated from

init-aggression-mean is not the one the spider will ultimately get if *aggression-distribution* is set to binary or bimodal -- see initialization for more details

- *aggression-sd*, the standard deviation used for generating aggression values for both the initial population and further generations, unless the *aggression-distribution* is set to binary (in which case the sd is overridden to zero). Can range between 0 and 100, and is left at 10 by default.
- *prey-abundance*, a number between 0 and 1. Defines the chance that a prey will spawn each tick. By default this is set to 0.25
- *inquiline-abundance*, a number between 0 and 1. Defines the chance that an inquiline will spawn each tick. By default this is set to 0.05
- *death-by-age?*, a boolean. If true, spiders have a 1-in-20 chance of dying each tick once their age is greater than time-to-elderhood. By default this is set to true.

There are many more parameters in the code, most of which have no variable name and are defined on-the-fly. Here I will only list those of that are mentioned elsewhere in this paper:

- *web-fray-rate*, which determines how quickly the webbing of patches decays. The smaller this number, the slower the decay. By default this is set to 0.1
- *base-prey-juiciness*, which determines how much energy a size 1 prey will give to whoever eats it. As all prey in the model are size 2 by default, you can just double this number to know the payout. By default this is set to 100.
- *metabolism*, which defines how much energy spiders and inquilines lose per tick. For spiders that are currently doing nothing, this number is halved. By default this is set to 1.
- *time-to-elderhood*, which defines how long a spider can live for before it starts risking a 1-in-20 chance of death each tick (assuming *death-by-age?* is set to true). By default this is set to 200.

Out of all of the parameters in the model (whether stated explicitly as a variable or implicitly as a number), only *aggression-distribution, inquiline-abundance,* and *prey-abundance* will be varied in testing.

INITIALIZATION

This model runs a method called *setup* on initialization, which does the following:

```
to setup
  clear-all
  setup-globals
  setup-patches
  setup-agents
  reset-ticks
end
```

clear-all and *reset-ticks* should hopefully be self-explanatory.

setup-globals defines all global variables that are not already defined in the interface, such as *edges*, *food*, and *max-turtles*.

setup-patches initializes the webbing value of all patches to 0 by default.

setup-agents defines the default shapes for the three agent types (spiders, prey, inquilines), then creates the initial population of spiders. When creating this initial population, the following procedure is done to determine the aggression score for each individual:

Create two local variables, aggression-mean and sd, and define them as *init-mean-aggression* and *aggression-sd* accordingly.

If *aggression-distribution* is not set to normal, then a random number is rolled from a normal distribution with aggression-mean as the mean and sd as the standard deviation.

If *aggression-distribution* is binary or normal: if the rolled number is less than 5, then set that spider's aggression-mean to 5. Otherwise, set it to 95.

If *aggression-distribution* is flat: if the rolled number is less than 5, then roll a number between 0 and 50, and assign that new number to aggression-mean. Otherwise, roll between 50 and 100, and set aggression-mean to that number.

If *aggression-distribution* is binary or flat, then set sd to 0

Finally, take the current aggression-mean and sd and use them to generate a random number from a normal distribution. That random number gets capped between 0 and 100 (if it isn't already) and assigned to the spider's *aggression* value.

Spiders that are bred later on use their mother's aggression score in place of init-mean-aggression and skip the second step entirely, but otherwise follow this same procedure for generating their aggression score.

DYNAMICS

The main method for running and controlling the dynamics of the model is *go*:

```
to go
  if not any? spiders [stop]
  if count turtles < max-turtles [env-spawn]
  update-prey
  update-inquilines
  update-spiders
  update-webbing
  tick
end
```

Pictured: a clown car

The first line of this method ensures that the simulation will end if the colony goes extinct, as for our purposes there is no point to letting it continue past that point.

The second line allows for new inquilines and prey to be added to the simulation if it isn't already overcrowded. Up to one prey and one inquiline can be spawned each tick.

update-prey and *update-inquilines* are responsible for controlling and updating their respective agent types. They aren't our focus, so let's skip ahead to *update-spiders.*

First, *update-spiders* checks the energy level of every spider. If a spider has no energy left, it drops dead. If *death-by-age?* is enabled (which it is by default), the spider is old (*age* > *time-to-elderhood*), and they roll badly, they die and donate their energy to the colony *food* bank.

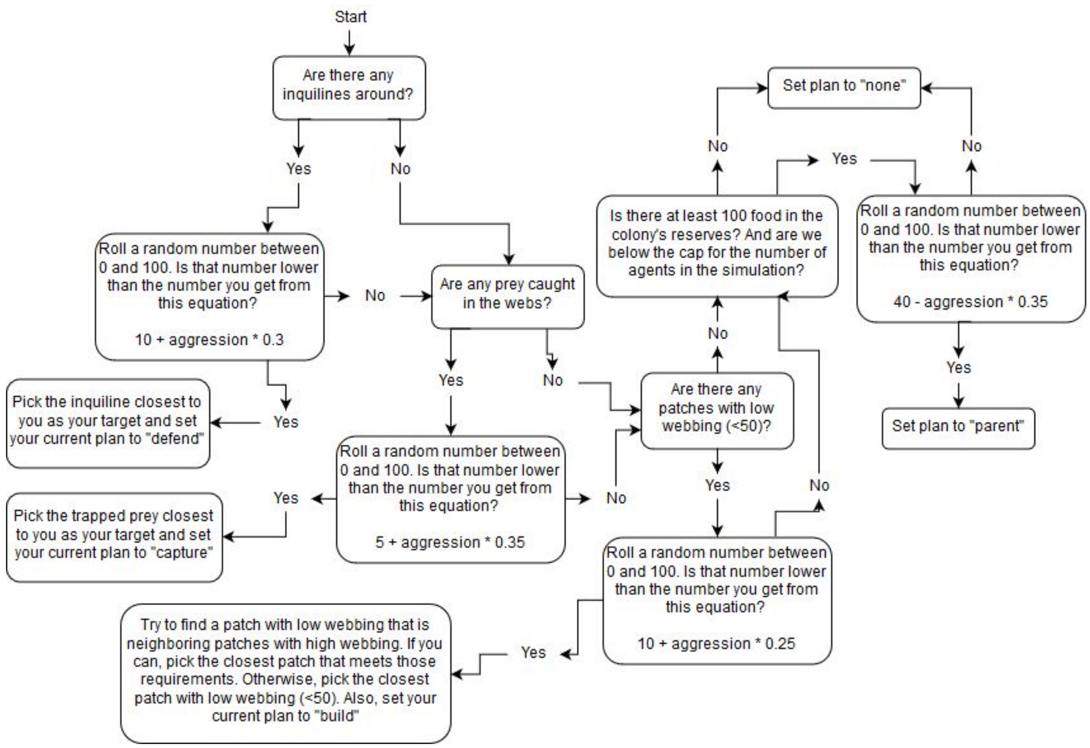
Then the spiders run *update-plan* and *enact-plan*, in that order. We will return to them shortly, for now let's go through the rest of *update-spiders*.

All spiders have their *energy* deducted by *metabolism* unless the spider's current plan is "none", in which case energy is only deducted by half of *metabolism*. If at this point a spider's energy is below a certain threshold (15 by default), then the spider will take some food from the colony food bank (up to either the total amount of food or as much as it'd take to bring energy back up to 100, whichever is smaller) and add it to their energy. All spiders' age by one before *update-spiders* ends.

update-plan causes the spider running it to either *scheme* or try to find a new target if they meet certain conditions. If the spider has no plan currently, or is trying to parent while there are too many agents, then they will *scheme*. If they are trying to catch prey or defend the colony and their target is gone, they will run *find-new-target* with the set of stuck prey or the set of inquilines, respectively.

find-new-target is very simple: it takes an agentset as an argument, and checks to see if there is anything in that agentset. If so, the spider picks a random agent from that set to be their new target. Otherwise, they *scheme*.

So what does *scheme* do? It's the method that controls which state the spider running it will move to. It works like this:



The parameter values used in this flowchart are out of date, but I'm tired and don't wanna fix it

.

Once our spider has a plan ready, the next step is to *enact-plan*. This method calls different functions depending on what the spider's current plan is. Said functions are what drive the spider's behavior.

Once the spiders have finished all of their actions, the last thing to do is *update-webbing*. This method simply decays the webbing value of all patches by *web-fray-rate*% (0.1% by default) and recolors them appropriately. After that, the clock ticks and this whole process can start again from the beginning.

MEASURES

There are two different kinds of experiments that I will be running with the model to test how well it has met its two design goals. The first kind, labelled "Task Participation Frequency" in BehaviorSpace, was created for the purposes of searching for parameters that will elicit the same (or similar enough) frequencies of task participation among docile and aggressive spiders found in Wright, Holbrook, and Pruitt (2014).

Because Wright, Holbrook, and Pruitt performed their experiments carefully in laboratory conditions, this experiment runs a special method called *task-test* in lieu of *go*. task-test moves through four different stages, each lasting 100 ticks, in which an ideal situation is created to encourage the spiders to perform the relevant task and none others, without any of the risk of starvation or senescence. The proportion of docile and aggressive spiders doing the phase-appropriate task are recorded each tick, and at the end of the experiment the averages are reported. This test is done with binary and bimodal aggression distributions and a range of different random seeds, for a total of 8888 runs (gotta stick with the theme).

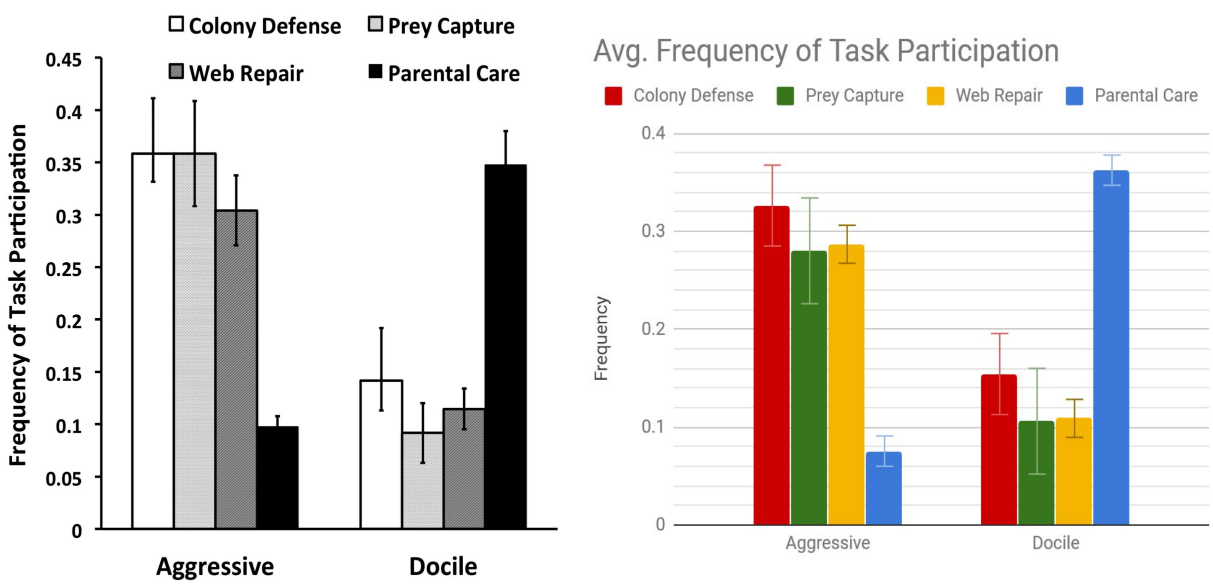
For testing the model for colony compositions in different environments (and different distributions of aggression), there are the “Colony Composition” experiments. These experiments collectively test all four different distributions of aggression, along with a range of *prey-abundance* and *inquiline-abundance* from 0 to 1 by steps of 0.5 and four random seeds (each tested to ensure a 1:1 aggressive:docile ratio for the initial population for their experiment). All other parameters are kept at their default values.

Each run of the experiment goes until either the colony dies out or the run reaches 400 ticks, which is long enough to (hopefully) ensure the first and second generations die out. At the end of the run the number of docile spiders, aggressive spiders, and spiders total are counted, along with the highest generation number reached.

Now, to compare our results to Pruitt and Goodnight (2014) we need to know the population sizes of our colonies, how many of those spiders are aggressive (as opposed to docile), and if the colony site is a “high” or “low” resource site. Unfortunately, the authors of this paper make no mention of how they evaluated whether a site had high or low amounts of resources. It seems like a safe assumption, however, that a plentiful site would have lots of prey and few competitors. As such, I will be using the prey and inquiline abundance parameters as a stand-in for resource abundance.

Results

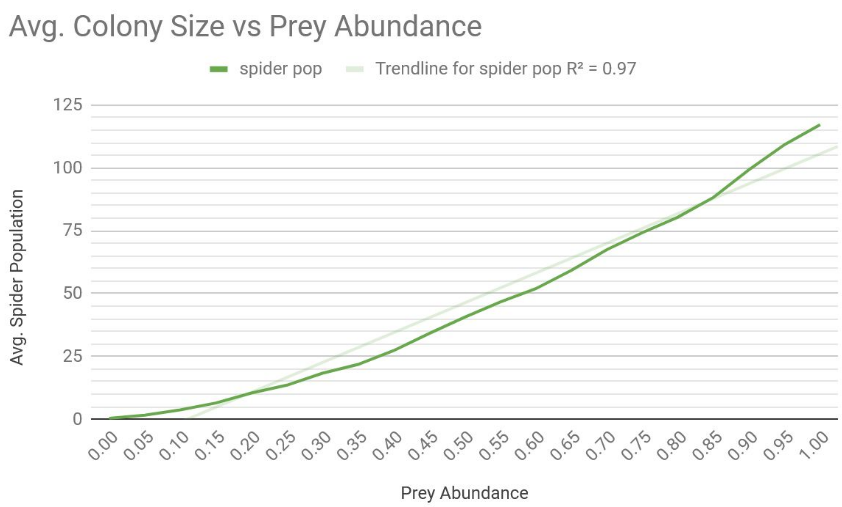
First, the results of task participation frequency test:

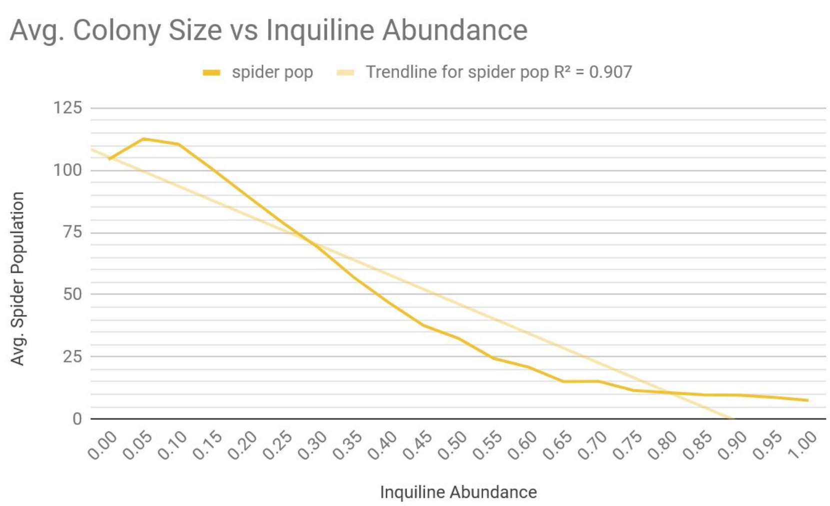


Left: A column chart of frequency of task participation from Holbrook, C., Pruitt, J., & Wright, C. (2014); Right: A column chart of the same from our last results of the task participation frequency experiment.

I wasn't able to get my results to match up to the original data as close as I'd like. That range and disparity on the prey capture task is especially worrying, and I'd have preferred to spend more time trying to find better parameters if I wasn't short on time. But as they say, perfect is the enemy of good, and there's more to do. If I or someone else can figure out how to get BehaviorSearch to play nice with this model later on then that could be a great boon to sorting this out in the future. If in the meantime you'd like to get your hands on this data and charts, you can go to this link: https://docs.google.com/spreadsheets/d/1IzFzF0BoeMvE2rEakIRnaHHFOy5zATVsUyWIaeZtd4Y/edit?usp=sharing

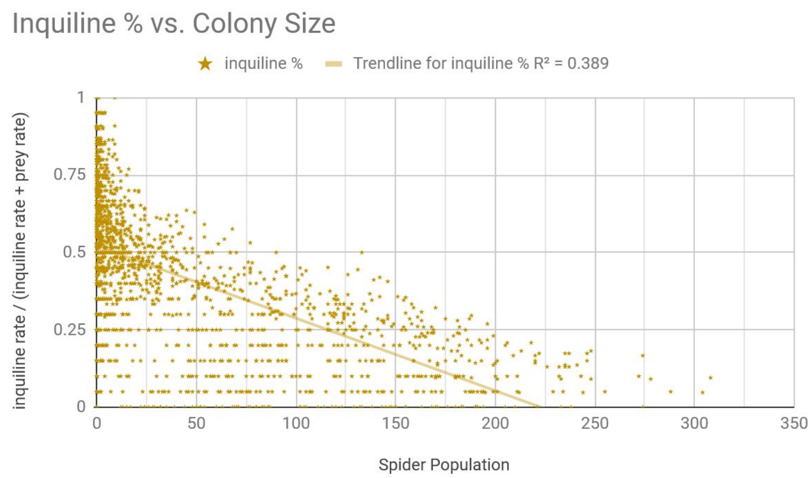
Once I got my parameters it was time to run the battery of colony composition tests. Now we can get more into the meat of this, starting with:



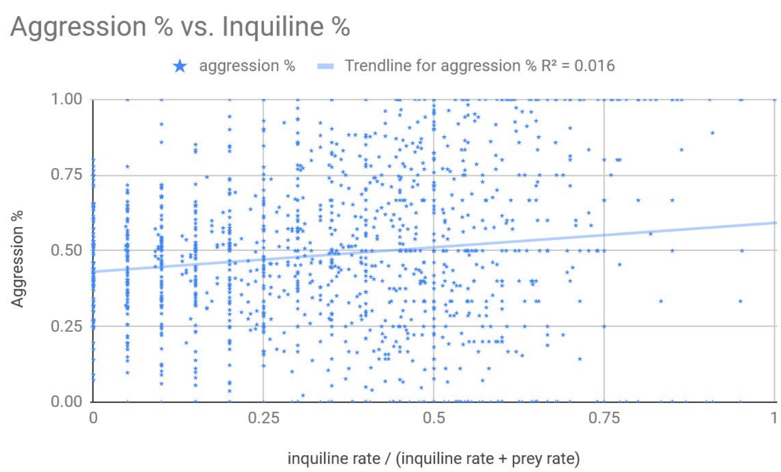


Nice, we got some strong correlations between colony population size and the spawn rates of different agents in the model! As expected, greater numbers of prey and lesser (but not no) inquilines are better for population growth.

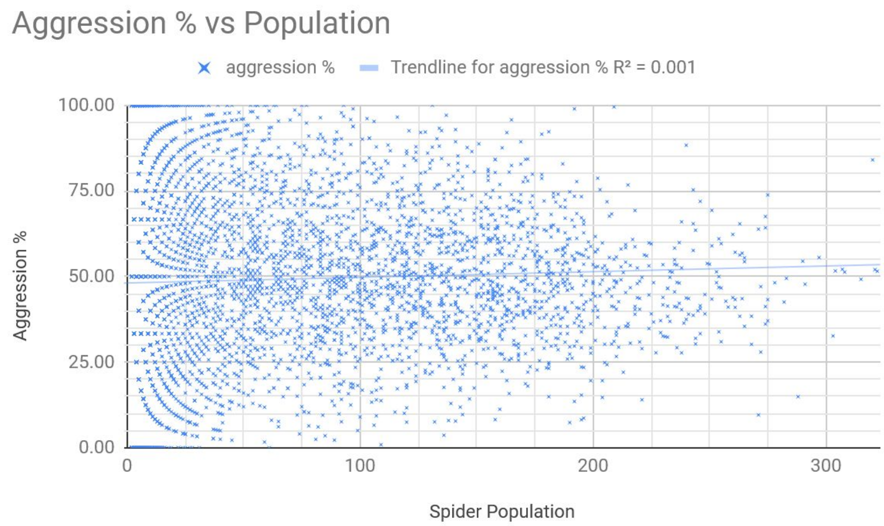
For the sake of clarity going forward, let’s assume that high resource sites would have lots of prey compared to inquilines, and that low resource sites have few prey compared to inquilines. We can quantify how highly valued a colony site is by dividing the inquiline abundance of the area by the sum of that abundance and the prey abundance to get a percentage. A high resource site would have a low “inquiline percentage” and vice versa. Let’s see how inquiline percentage correlates with population:



If you remember, Pruitt and Goodnight (2014) found that in high resource sites, small colonies have more docile than aggressive spiders and large colonies have more aggressive than docile, and vice-versa for low resource sites. Given that the “low resource sites” in this data all have small populations, we should expect to see that aggression increases with inquiline %. We did in fact see a little bit of that, but not as much as I’d have expected:

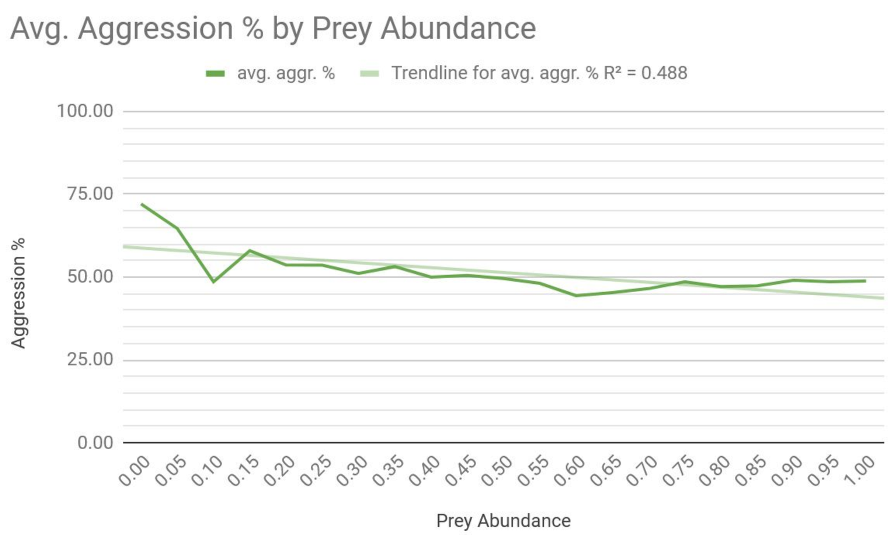


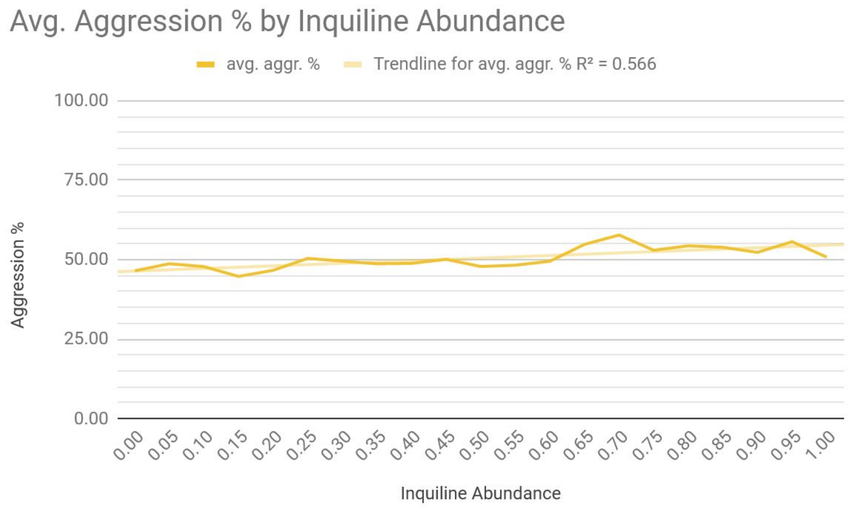
As you can see, there is a lot of scatter -- so much so that I’d likely not have noticed that there is that slight increase of aggression if not for that trendline! Not very promising at all. Also, what’s with those strange lines of data points in the sub-25% inquiline % range? Some similar kind of funk is also happening in the chart before it, and this graph comparing aggression and population size:



Higher populations tend to on average be more balanced than smaller colonies, which while very neat does not quite match up but Goodman and Pruitt (2014).

Finally, if we compare inquiline abundance or prey abundance on its own to the average of aggression:





We can see that there tends to be fewer aggressive spiders in colonies that have access to more prey and that (to a lesser extent) there are more aggressive spiders in areas with more competition, but again the correlations are too insignificant!

You might notice that in none of these charts have I compared different kind of personality distributions. It is not that I didn't do analysis for them, but rather that I found my results too lackluster to show for various reasons. Additionally, there is much more I could've done with my data in terms of analysis that I couldn't get to, either due to ignorance or a lack of time. But, if you are interested in looking at or playing around with the data and charts (including the ones I didn't show), then you can find them here: https://docs.google.com/spreadsheets/d/1ulDiR_mUvcJRl6cr3iwRpRVfb7TJIbvTqyHhf8HfRp8/edit?usp=sharing

Discussion

Given the results, I think it's safe to say that this model has failed to live up to being a replication of the real thing, at least as far as Pruitt and Goodnight (2014) are concerned. I feel that there may be a quite a number of factors involved as to why this project fell short, among them being:

- Insufficient testing. (ex. Lots of odd outliers in the data)
- Too many independent variables.
- Not enough statistical skill and know-how. (ex. I forgot how to calculate p and frankly it's going to take me too long to relearn how)
- Insufficient parameter tuning. (ex. See results of the task participation frequency test)
- Undetected bugs.
- Issues with the underlying theory or its implementation (ex. Should "parental care" really be a single task on-par with "defend colony" or "capture prey"?)

In spite of all that, I'm still very happy with the work I did on this project overall. I learned a lot and taught other people something new about spiders along the way, and I got a cool virtual spider terrarium out of the process! Even if the model is wrong in so many little ways, it already has value as a general educational demonstration and entertaining widget, and for that I can't help but be proud of my failure.

Acknowledgements

- Uri Wilensky, for the colorful histogram code from his model "Crystallization Moving", which can be found in the Netlogo Model Library under Sample Models > Chemistry & Physics > Crystallization
- Paul Smaldino and Matt Turner, for running the class this project was made for, for giving great advice, and for showing off some cool models and papers.
- My mother, for being an enthusiastic test subject and cheerleader.
- Jonathan Wojcik, for his *Spiderween* article on social spiders, which I saw first and subsequently felt inspired by. This article can be found at: http://www.bogleech.com/spiders/spiders22-social.html

References

- Pruitt, J., & Goodnight, C. (2014). Site-specific group selection drives locally adapted group compositions. *Nature, 514,* 359-362. https://doi.org/10.1038/nature13811. Retrieved from https://www.nature.com/articles/nature13811
- Wright, C., Holbrook, C., & Pruitt, J. (2014). Animal personality aligns task specialization and task proficiency in a spider society. *PNAS, 111 (26),* 95330-9537. https://doi.org/10.1073/pnas.1400850111. Retrieved from https://www.pnas.org/content/111/26/9533