

DSA Final Report

Jeremy Skoler, Katie Foster

May 2020

1 Question

There are many complex factors at work in an ecosystem, with different species impacting each other's populations. As a result, it can be difficult to figure out which species to allocate conservation money to so that the entire ecosystem benefits the most. There are many variables that need to be considered in order to make a decision. One is biodiversity, which refers to how different species are from each other. The distribution of money should aim to maximize the amount of biodiversity that survives. Another factor is how valuable a species is in other ways. A keystone species is a species which, if removed, would cause the entire ecosystem to collapse. For example, the beaver in North America makes dams, which enable many other species to thrive. Even if a species is not a keystone species, if it is the main source of food for other species, extinction of that one species would cause the extinction of all the other species that depend on it. Another important aspect to consider is how much the money will help each animal. It might take a million dollars to save one endangered species from extinction, but only a thousand to save another species.

2 Method

We decided to create an algorithm that would allocate money to different species by optimizing for value and biodiversity given a set budget.

First we created a class called `Animal`. Each `Animal` has a name, survival parameters, and a value constant. The survival parameters consisted of a list of three numbers. Those numbers were parameters a , b , and c in the following equation. We used it within the `Animal` class function "get-survival-rate" to calculate the chance of the animal surviving when given a certain amount of resources.

$$a\sqrt{resources/b} + c$$

We chose a square root function, because this function has a high slope at the beginning, but levels out as resources increase. The parameters are different for each species, since each species has a different chance of survival without

resources, and a different slope for how much resources would be helpful to them. This method allowed us to account for these differences.

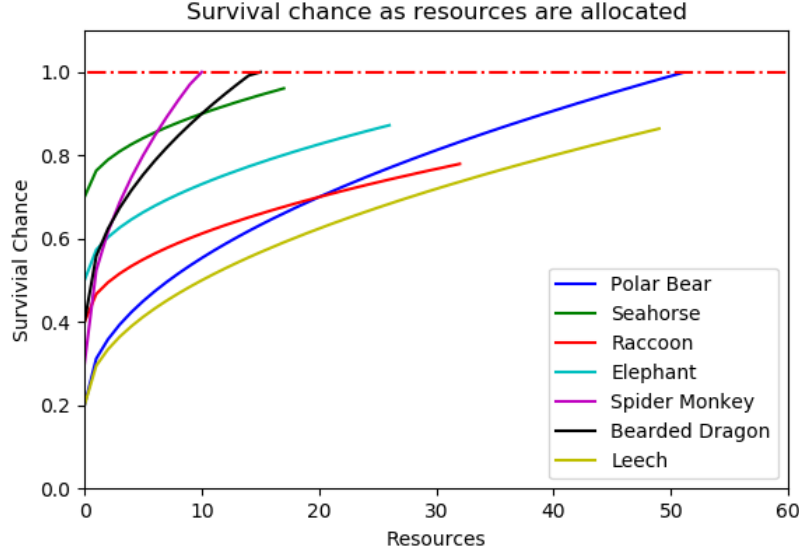


Figure 1: Graph of chance of survival vs resources allocated for different animals

As the graph shows, some species reach a survival chance of 1 before all the resources are allocated. As we are dealing with probability, any number over 1 is invalid, as an event can never have more than a 100 percent chance of occurring. To account for this behavior, our algorithm will return 1 for any survival chance over one, which effectively makes the equation a piece wise function that never exceeds 1.

Once the survival chance has been calculated, the Animal class method “get-value” multiplies the survival chance by the value constant to produce the overall potential value of the animal, given the allocated resources. The algorithm attempts to maximize the total potential value of the ecosystem while also maximizing biodiversity.

The main part of the algorithm is a function called resource-allocation. Resource-allocation uses a dynamic programming approach to try to maximize the ecosystem value by iterating through possible distributions of resources among animals. The ecosystem value is dependent on the “t-value”, which is the sum over all animals of each animal’s (value*chance of survival). The ecosystem value also takes into account the biodiversity, which is the product of all the animals’ survival rates.

The algorithm starts by finding the sum of all the animals’ values if they are not given any resources. Then, it calculates the biodiversity by finding the product of all the animal’s survival rates. Then the algorithm multiplies these

two values to find the ecosystem value, which is then added to the first row of the newly created ecosystem matrix, (size: number of resources * number of animals).

Then the algorithm enters a loop in which it goes through every resource and determines where to allocate it. For the first animal, it assumes that the resource is initially allocated to the first animal in the list, and then updates the ecosystem matrix assuming allocation. Then we enter a loop that goes through each animal. For each animal, the algorithm calculates the potential total value and potential biodiversity value, and uses these values to calculate the potential ecosystem value.

```

1 pot_Tvalue = last_round_Tval + (a.get_value(a.resources+1)-a.
  get_value(a.resources))
2 pot_bio = (biodiversity/a.get_survival_rate(a.resources)) * a.
  get_survival_rate(a.resources+1)
3 pot_eco_val = pot_Tvalue * pot_bio

```

Then there is an if statement which checks if the potential ecosystem value is better than the last best one, and if so, updates the value. This loop will find the most beneficial animal to assign a resource to (the “lucky duck”), and keep track of that animal. Once the loop ends, the animal who receives the resource has their number of resources incremented, and the biodiversity value is also updated.

Once every resource has been assigned, the algorithm formats and returns the results.

3 Results

Although we were unfortunately unable to find any real life data to verify our algorithm, it performed well on the made up data that we fed into it.

First, we did a test using just three animals. These animals were a fish (which had a very high value and chance of survival), and a penguin and seal. The penguin and seal both had the same chance of survival, but the penguin had a slightly higher value of 3, while the seal had a value of 2.

```

1 fish = Animal('fish', [0.2, 5, 0.9] , 8)
2 penguin = Animal('penguin', [0.4, 20, 0.1] , 3)
3 seal = Animal('seal', [0.4, 20, 0.1] , 2)

```

From this setup, we expected the fish to receive few if any resources, and for the penguin to receive more resources than the seal. We called the algorithm with a resource number of 3, to test a simple case. The results below are just as we had expected. Although the fish was not allocated any resources, it has a 90 percent baseline chance of survival, so it does not need resources as much as the other two species.

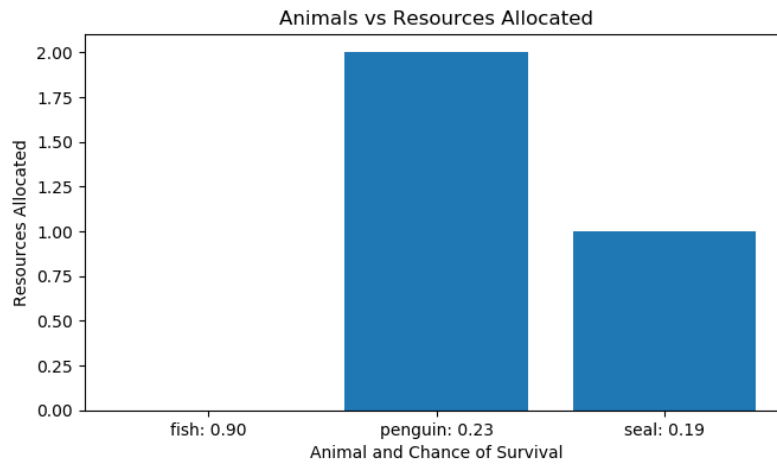


Figure 2: Results for test using three Antarctic animals

Next we ran a test on a list of six animals, using a resource amount of 100.

```

1 cow = Animal ('cow', [0.5, 20, 0.5], 8)
2 cat = Animal ('cat', [0.1, 10, 0.7], 5)
3 mouse = Animal ('mouse', [0.3, 20, 0.4], 7)
4 elephant = Animal ('elephant', [0.5, 30, 0.1], 4)
5 horse = Animal ('horse', [1, 20, 0.3], 2)
6 beardedDragon = Animal('bDragon', [0.5, 10, 0.4], 2)

```

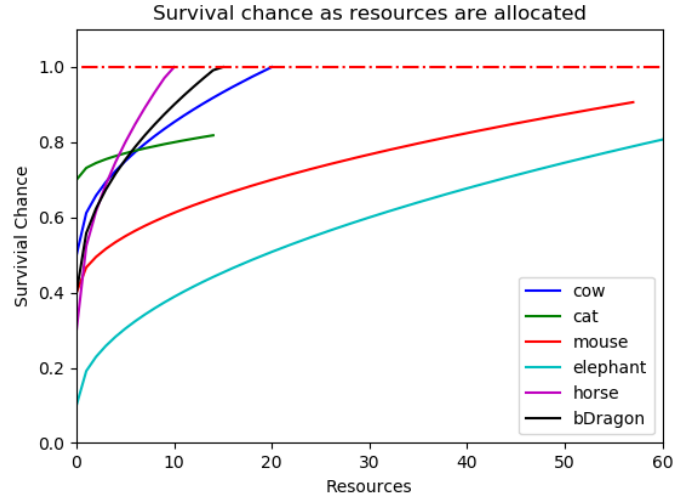


Figure 3: Visualization of survival curves and an allocation of resources for each animal

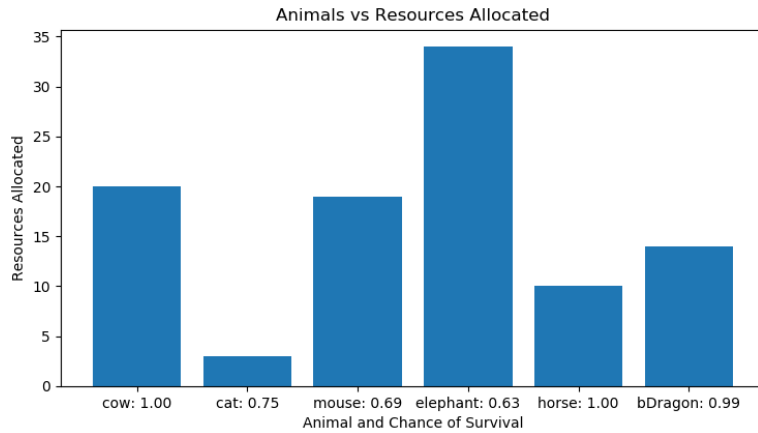


Figure 4: Results for test with six animals

Overall, the algorithm worked as we expected. The algorithm allocated resources with a preference towards the animals with high value and/or low initial chance of survival, but also avoided allocating resources to just one or two animals.

We wrote a function to simulate the algorithm being used in real life. The simulation function went through each animal and used the chance that the animal survived to make a random choice of whether it survived or not.

```

1 # find out if animal survives (random)
2 animal_survives = random.random() <= prob
3 if animal_survives:
4     value += a.value_constant # add animal's value to total value

```

After going through each animal, the algorithm returned the total value. When we ran it on the example above with six animals, the algorithm had good results. We knew the minimum value if no animals were given resources was 12, and the value if all the animals survived was 28. Our algorithm had limited resources, and was not able to bring all animals to a 100 percent survival rate. However, it ended up giving us a simulated value of 22.9 when ran 500 iterations assuming 100 resources were available. This is 82 percent of what the total value would be if all animals survived.

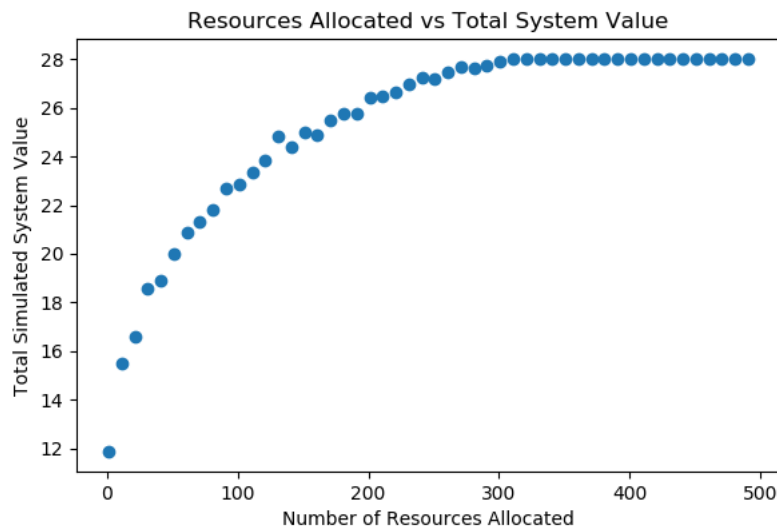


Figure 5: Resources vs value in simulation

We created Figure 5 by varying the number of resources that the algorithm could use, and then running the simulation function. The result was a graph that showed how the resources allocated would affect the total value of the population. The graph shows that resources make more of a difference at the beginning, and that this particular system requires about 300 resources to consistently have all the animals survive.

4 Interpretation

In order to make this algorithm, we were forced to make several assumptions. As we are not biologists, many of these assumptions were not backed by evidence, and were simply our best guess. One of the things we assumed was the value and response to intervention of a species. Ecosystems and even individual species have so many different factors influencing their value and survival, that it is impossible to come up with an exact and correct estimate for their value to the environment, chance of survival, or response to intervention. Another assumption was using the square root function with parameters to model the species response to aid. We based that function on the economic concept of diminishing returns, but we are not sure if this is actually how the real curve would appear. It might look like an s curve, or it might be linear. The final assumption was that each animal has a value independent from other animals. Our model tried to preserve the most biodiversity, but it did not account for the value of one species to the environment might decrease if another species dies. For example, Koalas eat only Eucalyptus leaves, so while Koalas survive, the Eucalyptus plant is very valuable, but if they were to die, the plant would have slightly less value to the ecosystem.

Using an algorithm like this one to decide where money is allocated in a real world scenario would have a big impact. Hopefully, it would allow people to help the ecosystem the maximum amount that they are able to. However, a model is only as good as the assumptions that are built into it, and this model has several assumptions that are not backed by evidence. Therefore, this model should be reviewed by biologists and tested with real world data before it is used in a real high stakes scenario.