Yazan Bawaqna

CS 152

Project 6

November 2nd 2022

**Darted elephant’s optimization for elephant population in Kruger National Park using python**

**Abstract:**

As a continuation of the previous project, this project continues to program function for the elephant population in Kruger National Park, South Africa. In this project, we will be focusing on using a search method that will allow us to find the optimal parameters for the percentage of elephants getting darted. The search type we will use is called binary search, which works faster than the standard linear search. While the linear search goes over the data one by one to find the required data, the binary search sorts the data and goes to the mid-section of the data and is able to decide if the required data is greater or less than the mid-data. Using this method, binary search keeps cutting the data in half and excluding 50% of the remaining options for every search. This method of search increases the program’s efficiency as it reduces the search time. In programming, we call the run time of a program time complexity, which means the less the time complexity the faster and more efficient the program is. Hence, binary search is a way of reducing time complexity.

Besides the usage of binary search and optimization, the program also uses nested loops and multi-dimensional lists at its core. Multidimensional lists help keep the code tidy and neat by storing the data under one multi-dimensional list. This can be seen when storing the individual elephant’s characteristics list inside the total elephant demographic list. Furthermore, nested loops allow for a faster method of performing the function than doing it manually.

**Results:**

**Here we can find the five required output graphs drawn by matplot via python**

1. **Vary the adult survival probability from 0.98 to 1.0 in steps of 0.001.(example above)**

Chart, scatter chart

Description automatically generated

We can see that the optimal percent darted is increasing linearly as expected

The adult survival probability grows from 0.98 to an astonishing 0.9975

**Vary the calf survival probability from 0.80 to 0.90 in steps of 0.01.**

**Chart, scatter chart

Description automatically generated**

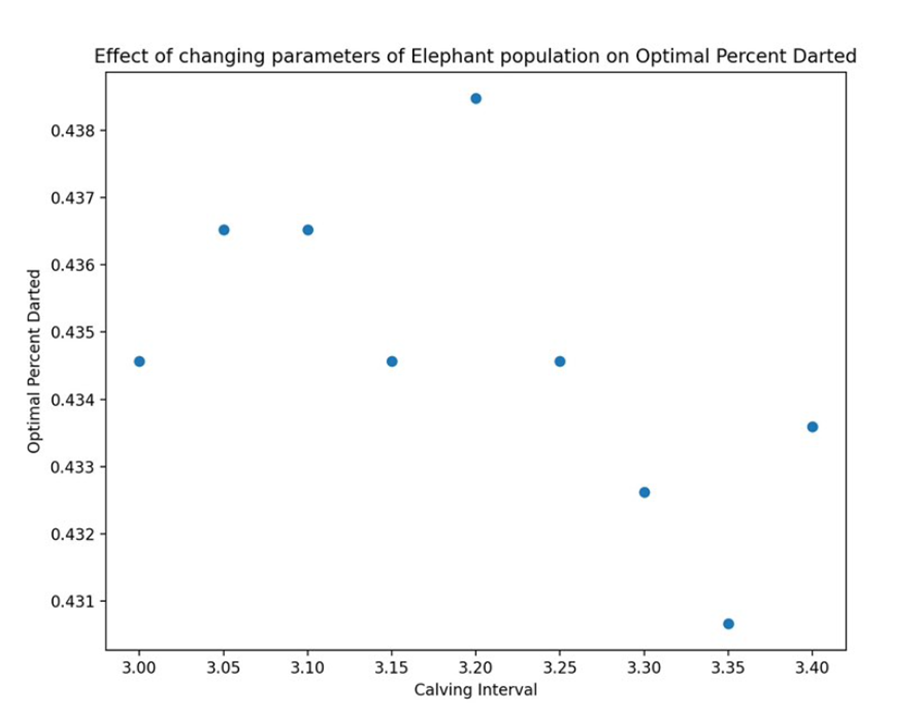
**3.Vary the senior survival probability from 0.1 to 0.5 in steps of 0.05.**

Graphical user interface, application

Description automatically generated

We can see that for the senior survival probability the correlation with the optimal percent darted, that maybe due to the fact that senior elephants are prone to multiple disease causes which are harder to track. However there is a considerable positive correlation of we excluded the second data point.

1. **Vary the calving interval from 3.0 to 3.4 in steps of 0.05.**

****

**Very weak negative correlation. Even though the points do not align for a line plot, there is a negative relationship to be considered**

**5. Vary the max age from 56 to 66 in steps of 2.**

Graphical user interface, application

Description automatically generated

**Strong positive correlation**

**Follow up discussion:**

I read a book about the advancement in genetic technology, and I remembered that it mentioned the concept of optimization, I did not really understand the concept back then, but reflecting on that now makes me understand how this concept works. After I did some research today, I understood that genes can be interpreted as numbers by giving their chromosomes values. Since genes are primarily a collection of chromosomes, a list can be made to collect each gene’s chromosomes. Using “random” which can be imported from the python library, we can create the genes of the simulated population in accordance with the natural probability. Next, is where the optimization comes in place, different gene compositions create different survival fitness for different environments. Hence, as explained above optimization can be used for the search for the most surviving genes. This has the power of simulating animals in danger or natural/artificial selection simulations

**Follow-up questions:**

1. **What does an import statement do?**

In python, the import command extracts the module of code from the “imported file/module” into the current program.

1. **What is binary search? How does it work differently than a linear search algorithm?**

The linear search goes over the list of data sequentially one by one until it finds the target data. However, binary is more efficient as it goes to the middle of the data and keeps cutting the data from the middle until the middle element is the target element

1. **Why is binary search faster than a linear search (e.g. going page by page to find a word in a dictionary)?**

Simply because with binary search the search will not go threw every element in the data list

1. **For full credit, the data must be interpreted for the reader in clear sentences within the writeup. What do these numbers mean? Why do the darting percentage go up/down when you vary each parameter? How should Kruger National Park use this information**?
2. **How might you apply this type of optimized search algorithm approach to something you are really interested in?**

Since I am primarily interested in math, python optimization with binary search can be used for polynomial equations optimization by optimizing the parameters. Those mathematical equations can be used for engineering designs or profit margin calculations

**Acknowledgements:**

Kalyan, fixed my optimization code.

David Istrati, debugged my project 5 issues.

Magzhan tressiov, done the lab with me, helped me finish it quicker.

Pramithus, made my plot more clear and accurate.

<https://towardsdatascience.com/genetic-algorithm-implementation-in-python-5ab67bb124a6>, genes information

Sean helped me draw the grapjs