Search for a car parked on a forest road

Final Report: May 6th, 2014

Math 485

Group Members:
Mohammed Alabkari, Rachel Anne Braun, Yu Gong, Alex Nguyen, Jayson Anthony Rickel

Mentor:
Gleb Zhelezov

Introduction

Mathematical modeling is a very useful and helpful tool for studying the search for a parked car. Imagine parking on the side of a forest road in order to go hiking. After hiking through the woods for a while, you emerge from the trees and are once again on the road. However, you are no longer near your car. You can assume that you know the probability density for the location of the car, since you didn't hike too far. What is the best way to go about finding your car? Through mathematical modeling, we can discover an optimal search pattern for finding the parked car.

In the traditional linear search problem, an object is placed on a line using a given probability density. Starting with an initial step length, the searcher goes back and forth using step lengths calculated from a specific formula until the object is found. To simplify our analysis, we consider the one-sided search problem (Figure 1). Applications of such a problem can include a simple robot that returns to the origin after each step to report whether it has found the car or not.

The overall goal of our project is the analysis of a linear search problem for an object placed using a given probability density function using analytical and numerical methods. Currently, we have developed numerical methods simulating the search for an object placed using an exponential distribution function and the minimization of the search for the object. Using Matlab, we have modeled an assortment of searches to determine an optimal sequence for our fixed parameters.

In order to increase the complexity of our model and model a more realistic problem, we have introduced a further sight parameter. This parameter allows the searcher to see an additional distance ahead after each step. Applications of this problem include a simple robot on a winding road fitted with a camera. After each step, the robot can look ahead a different distance, depending on the curves in the road, and determine whether the car is there or not. Using Matlab and Java, we have modeled the searches using a sight function determined by a normal distribution. Again, the optimal stepping pattern was determined for this situation. Furthermore, the effects of the sight parameter on the total expectation length and the parameters used to determine step length were tested.

Through mathematical modeling, we can reduce the cost and time spent determining optimal paths that can later be applied in real-world situations. For example, we can test multiple probability distributions for the location of the car without physically setting up the parked car. In our project, mathematical models can help to predict the position of a car and further serve as guidelines for travelers and primary sources for researchers who want to observe the visualize aspects of each case on each experiment. For researchers, models provide an overview for a case that may not be practical in the real world. In summary, development of a model for the search for a parked car helps us to predict the car position, minimize the cost, and provide insight into real world behavior.

Mathematical Model

The following assumptions were made in our reference paper (Ref. 1):

The search problem is only on a one-sided gatherer version and the hidden object H, the car position, is located on the *half-line* $\mathbb{R}_+$. In our analysis, we consider the same one-sided searcher problem.

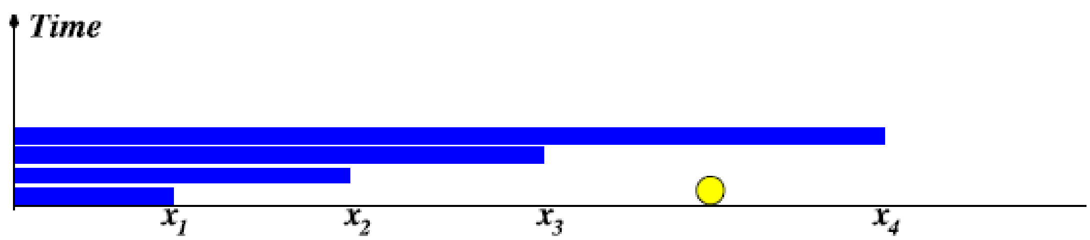

Figure 1. Diagram of one-sided gatherer movement (Ref 1.)

In the paper, the distribution function used was the homogeneous tail distribution function, also called a Pareto distribution, shown below.

$$x = \{0 = x_0 < x_1 < x_2 < \cdots < x_k < \cdots\}$$

However, in our analysis, we will use an exponential probability density for the car's location.

Using the Pareto distribution, the paper (Ref. 1) arrived at the following conclusion. The optimal plan should satisfy a two-term recurrence, the variational recursion.

$$f(x_{k-2}) + f'(x_{k-1})x_k = 0$$

$$x_k = \frac{f(x_{k-2})}{-f'(x_{k-1})}$$

$$E(x) = \mathbb{E}[L(x,H)] = \mathbb{E}\left[\sum_{k=1}^{n(x,H)} x_k\right]$$

However, in the theory section below, we analyze the problem from the perspective of a general given probability density function.

The following variables are used in our analysis:
x = step length, given by the geometric series: $x = \Delta\alpha^n$
Δ,α = geometric constants, to be optimized for minimization of search length
L(x,H): the total distance travelled until the point H is found, given as a function of x
$E(x) = \mathbb{E}L(x,H)$: the cost/expectation of the search plan as a function of the total distance travelled
P(x): chosen probability density function
p(x): cumulative distribution function for the chosen probability density function
S = sight parameter, determined by the following distribution:

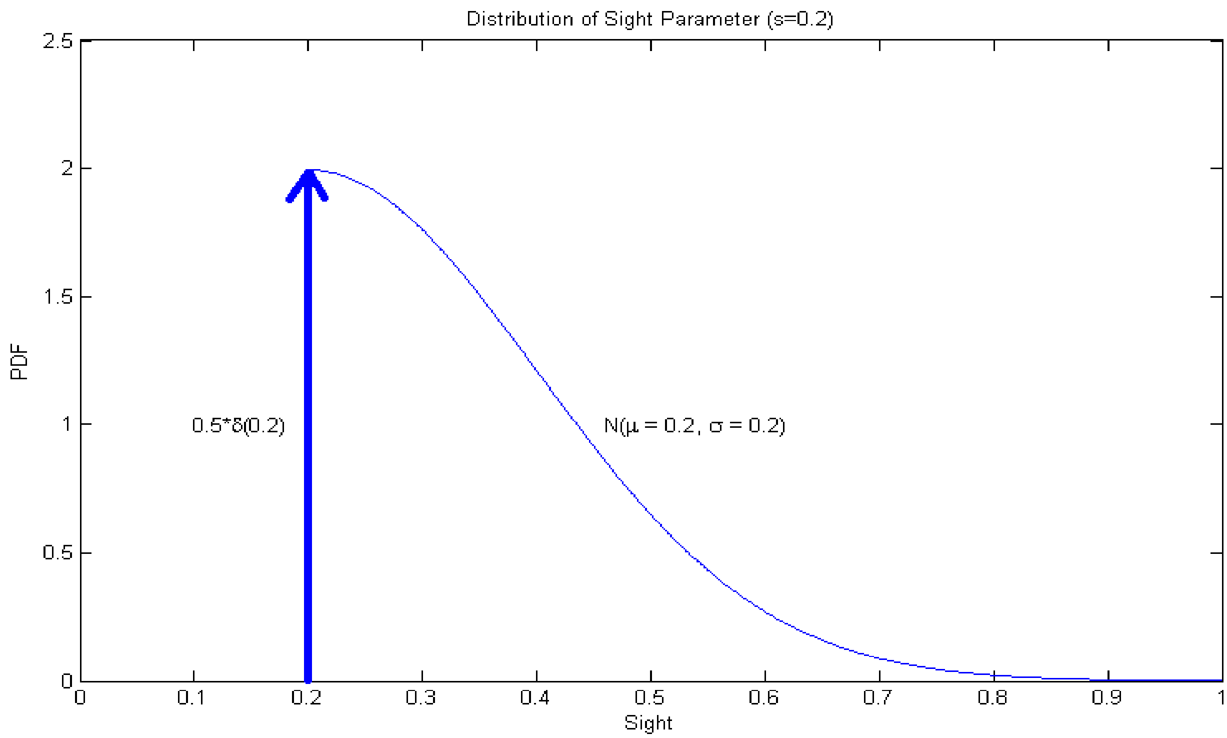

Figure 2. Normal distribution for sight parameter, with μ = .2, σ = .2

Theory

The following presents an analysis for the classic linear search problem.

I. Symmetry of the Problem

Starting at any point on the road and using the assumed probability density function, we would generally expect the result of a symmetric distribution for the searching process. For a searching process started by the center, we would go through each opposite direction with an equal search distance. The above two identities would have allowed us to only solve either side for the optimal result, typically as we are aiming at solving for the shortest value of the expectation for the cost function.

II. Approach to an Analytical Solution

There are however, two methods to approach the solution for the expectation for the cost function.

The first method is done by assuming a certain position of the object, solving for the expectation value for each single position, and then summing up all individual values by the end. Such a method is detailed in the following:

Here we define the probability density and its anti-derivative:

$$\frac{dp}{dx} = P(x) \qquad \text{Eq. 1}$$

Where P(x) is the probability density function and p(x) is the anti-derivative of probability density function, also known as the cumulative distribution function.

For each certain assumed position of the object with respect to the stepper index n, we shall have the individual expectation value such that:

$$\mathbb{E}\, L\,(n, x_1, x_2) = 2\int_0^{x_n} L \cdot P(x)dx \qquad \text{Eq. 2}$$

Where L is the total search distance for the single process:

$$L\,(n, x_1, x_2) = 2\sum_{n=1}^{N} x_n \qquad \text{Eq. 3}$$

Consequently, substituting L for the sum of each single step, we shall have the actual form of the summation as follows:

$$\mathbb{E}\, L\,(n, x_1, x_2) = 2x_1\int_0^{x_1} P(x)dx + 2x_2\int_0^{x_2} P(x)dx + \cdots + 2x_n\int_0^{x_n} P(x)dx = 2\sum_{n=1}^{N} x_n \cdot \int_0^{x_n} P(x)dx \qquad \text{Eq. 4}$$

For a final sum of all possible positions of the object, one shall rearrange the above expression in a better form such that:

$$\mathbb{E}\, L\,(n, x_1, x_2) = 2\sum_{n=1}^{N} x_n \cdot \int_0^{x_n} P(x)dx = 2\sum_{n=1}^{N} x_n \cdot [p(x_n) - p(0)] = 2\sum_{n=1}^{N} x_n \cdot p(x_n) - 2p(0) \cdot \sum_{n=1}^{N} x_n \qquad \text{Eq. 5}$$

The above equation is not yet the eventual answer we are looking for, since we need to assume the following equation for all the individual positions for the object. Therefore, we would need to evaluate each single value for the above expression with respect to different possible positions for the object and sum the results to find the eventual value for the final expectation of the cost function. From here, we could hopefully solve for the conditions for the minimization.

The above evaluation may seem to be a straight forward approach for solving the problem. However, one obvious disadvantage is that it takes many steps for the summation expression, which eventually arrives at yet another summation within the above summation. Nonetheless, there is not yet a better arrangement for a single summation term for the expression for the cost function. Consequently, a better approach is required for the further calculations.

The second method for approaching the problem is generally tricky but still a better method. We assume the potential position of the object is lying in between each step with respect to their probabilities of actually being in the position.

Therefore, the total distance for the searching process, using the condition that the object is lying in between each searching step, is listed as follows, respectively:

$$L_{1,2} = 2x_1\ , L_{2,3} = 2(x_1 + x_2)\ , \dots\ L_{n,n+1} = 2(x_1 + x_2 + \cdots + x_{n+1}) \quad \text{Eq. 6}$$

In a brief conclusion:

$$L_n = 2\sum_{n=1}^{N} x_n \qquad \text{Eq. 7}$$

Now, to treat the probability of the car lying in between the n-1 and nth step is relevantly regarding the probability of taking the total searching distance as long as L_n indicates. Therefore, a whole new rearrangement should be introduced to the solution approach:

$$\mathbb{E}\, L\,(n, x_1, x_2) = \int_0^{x_1} P(x)dx \cdot 2x_1 + \int_{x_1}^{x_2} P(x)dx \cdot 2(x_1 + x_2) + \cdots$$
$$\dots + \int_{x_1}^{x_2} P(x)dx \cdot 2\sum_{n=1}^{N} x_n \qquad \text{Eq. 8}$$

Hopefully, by the earlier definition of the anti-derivative of the probability density function (cumulative distribution function), we are able to solve the integral of each term in order to have a better rearrangement of the expression for the cost function:

$$\int_{x_{n-1}}^{x_n} P(x)dx = p(x_n) - p(x_{n-1}) \qquad \text{Eq. 9}$$

In the meantime, we could break the terms separately, and pair them in the following manner:

$$\mathbb{E}\, L\,(n, x_1, x_2) = 2\sum_{m=1}^{M}\int_{x_{n-1}}^{x_n} P(x)dx \cdot \sum_{n=1}^{N} x_n = 2\{x_1 \cdot [p(x_1) - p(x_2)] + x_1 \cdot [p(x_2) - p(x_1)] + x_2 \cdot [p(x_2) - p(x_1)] + \cdots x_{n-1} \cdot [p(x_{n+1}) - p(x_n) + x_n \cdot [p(x_{n+1}) - p(x_n)] + \cdots\} \qquad \text{Eq. 10}$$

For each sum of pairs of positive and negative $x_n \cdot p(x_n)$, the cancellation is thus available for driving the eventual form of the expression in terms of:

$$\mathbb{E}\, L\,(n, x_1, x_2) = 2 \cdot \sum_{n=1}^{\infty} x_n - 2 \cdot \sum_{n'=2}^{\infty} x_{n\prime} \cdot p(x_{n'-1}) \qquad \text{Eq. 11}$$

III. Analysis of Analytical Solution

As Eq. 11 indicates, there are two summations for infinitely many terms; yet as the probability density function can be randomly chosen, there is not much more simplification of Eq. 11 that can be performed. Therefore, we cannot solve the problem analytically past this point and instead must turn to numerical methods. However, one typical characteristic of the eventual expression would have granted us the possibility of solving for the minimizing condition with computational simulations. In the final Eq. 11, the expectation length is described as the difference between two values. Since the value of the probability function is less than one in nature, the function will converge to a certain value for the infinity sums.

IV. Introduction of a sight parameter

In practice, a sight parameter is required to further realize the modeling. The sight parameter itself must have the physical affections to the construction of the equation that the searching process is to be done whenever the object is within the visible range. To describe the visible range after each step, we decided to choose another governing probability density function instead of introducing a fixed length of the visible sight. One obvious reason for such a set-up is the assumption of non-uniform terrain along the searching process. We therefore, define the probability density for the visible range, say, the sight function as follows:

$$S(x_n) = \begin{cases} \frac{1}{2}\delta(x_n - \mu), \ x = \mu \\ \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x_n-\mu)^2}{2\sigma^2}}, \quad x > \mu; x < \mu \end{cases} \qquad \text{Eq. 12}$$

The above sight density function represents the probability of the actual length along the searching process, with respect to the position of the detector. We can see, however, several consequents result from the new definition which would cause extremely problematic conditions for the further analysis:

1. Overlapping

   By the looks of the probability density function, a minimum sight is ensured by the delta function in the first place; the vision, by chance, could be extremely long, which is capable of “seeing through” the whole distance to have the object spotted, thus ending the searching process. For instance, even by sitting at the origin of the searching process, the vision would still be extremely long, and could potentially spot the object before any steps were actually taken. These and other certain characteristics of the sight function can be quite consistent with reality, since we might expect to have the car spotted in the first place if the weather is not that bad, and the road is straightly long…etc.

2. Probability density function in nature
   As the sight density function is constructed in the first place, we wouldn't except to have any analytic solutions that fit in certain conditions as might be needed in further analysis. One for instance is the condition for if the object is spotted along the searching process is however, required for the evaluation of the expectation value for the total distance (the cost function), yet the probability density structure of the sight function provides quite an ambiguity for the solution.

A clear definition for the above condition is thus required. We hence introduce the “stopping time”, such that:

Let K be the first time that the object is spotted along the search road, such that: $x_K + s(x_K) \geq l$*, where:* $s(x_K)$ *is the probability function (anti-derivative of the sight function) of the sight, l is the assumed position for the object.*

The above definition offers us an increased chance for approaching the eventual results. Assumingly, the object lies solidly at the position l along the road, therefore, the chance of having the object found in Kth (the stopping time) step would be a product of the probability of object to be found in Kth step, and the probabilities of the object not to be found before Kth step, which is shown as follows:

$$p_k = p(x_1 : x_1 + s(x_1) < l) \cdot p(x_2 : x_2 + s(x_2) < l) \cdot \ldots \cdot p(x_{K-1} : x_{K-1} + s(x_{K-1}) < l) \cdot p(x_K : x_K + s(x_K) \geq l) \qquad \text{Eq. 13}$$

Where $p(x_n)$ is the assumed probability function for the object to be found with respect to the position of the detector along the road.

Therefore, p_k presents the individual probability for the object to be found until the Kth step of the searching process. So the total distance the detector has travelled is:

$$L_K = 2\sum_{n=1}^{K} x_n \qquad \text{Eq. 14}$$

Yet, the above probability function doesn't yield the whole probability of this saturation, since the object would only have a “chance” of randomly lying on l, we have to count the chance of such an occurrence:

$$individual\,probablity = \int_N^{N+1} f(x)\, p_k dx = \int_N^{N+1} f(x) \cdot \{p(x_1 : x_1 + s(x_1) < l) \cdot p(x_2 : x_2 + s(x_2) < l) \cdot \ldots \cdot p(x_{K-1} : x_{K-1} + s(x_{K-1}) < l) \cdot p(x_K : x_K + s(x_K) \geq l)\} \cdot dx \qquad \text{Eq. 15}$$

Where l is not necessarily lying merely near the Kth step, due to the overlapping, so that the above integral is assumed to be evaluated under the condition of the object lying in between Nth and N+1th step away from the Kth step. As we have already discussed before, the overlapping character of the sight function provides an extremely harsh condition for any further analysis.

The total distance we expect with respect to the probability density function and the sight density function is to be the sum of all those above terms from N=1 to infinity, and from K=1 to infinity, shown as follows:

$$\mathbb{E}L = \sum_{N=1}^{\infty} \sum_{K=1}^{\infty} \int_N^{N+1} f(x) \cdot p_{K \leq N} \cdot dx = \sum_{N=1}^{\infty} \sum_{K=1}^{\infty} \int_N^{N+1} f(x) \cdot \{p(x_1 : x_1 + s(x_1) < l) \cdot p(x_2 : x_2 + s(x_2) < l) \cdot \ldots + p(x_{K-1} : x_{K-1} + s(x_{K-1}) < l) \cdot p(x_K : x_K + s(x_K) \geq l)\} \cdot dxdx \qquad \text{Eq. 16}$$

The above equation may serve as an analytic result for our problem, yet it is impracticable for any simulation. We can see this result is impractical since by calculating the whole function, the “stopping time” must have a clear expression, so that the simulation could appreciate the process. However, as the probability density function for the sight in nature, the evaluation of the stopping time is almost impossible to execute.

Therefore, instead of the above conclusion, our calculation follows merely two simply logical processes:

1. Check at each position to see if the object is spotted. If so, then end the searching process.
2. If the object is not yet within the visible sight, then continue the searching process until condition 1 occurs.

It might be seemingly strange for such tedious calculations to result in the definition of only two logic process. However, the above logic process executes the evaluation process well in computational simulations. As detailed in the Numerical Methods and Computer Simulations section, we may find that hundreds or thousands of results from the simulation are chaotic and inconvincible, yet a million or a billion results would prove a general structure of the actual result of the searching process. Our previous analytical result has provided a strong milestone for the necessity of computational simulations to be involved.

Numerical Methods and Computer Simulations

We know that the infinite sum that defines the expectation of the length is infinite; however, we don't have an easy way of analytically computing the values of the first two step points which would recursively solve the entire pattern. This means we have to go to computer simulations in order to solve this problem; more specifically, we need to use the Monte Carlo Method. The Monte Carlo Method is a type of simulation that is used when there is a random event occurring in the simulation. For example, one can roll two, six-sided die over and over again and tally the results in order to find the probability of an individual roll begin rolled. In our case, our random events are the probability of the car being placed along the forest road and the sight parameter that is calculated at each stepping point. We need to run many (~one billion – one trillion) simulations of placing the car and finding the length to get to the car in order to optimize the step points that we want to use. We will pick two points to begin with, run the simulations, average the length to get to the car, move the points slightly, and continue this process until we find two step points that give us the minimum length to the car.

Before we do this, we want to see what sort of data the computer is dealing with to give us some insight into the behavior of this problem. We can run the simulation many times and create a histogram in order to see the curvature of the lengths that we receive by mixing the continuous probability density function with the piecewise function that represents the lengths for each car placement. In the paper, the probability density used is a parabolic density with exponent -α. This density gives an optimal pattern that is a geometric series. The density that we will be using is an exponential function with λ = 1 to simplify this function. However, unlike the parabolic equation, this density function will not give a perfectly geometric series of steps. We still know that the geometric series will give us a very good result, so we will be using this pattern to simplify these preliminary simulations. The geometric series will be represented in the following form: $x_n = \Delta\alpha^n$. The following figure shows us what the frequency of different lengths are when we set α = 1.10 and Δ = 0.30. This gives a nice curve that shows the mix of the geometric series that is seen to dominate the graph early and the exponential probability curve that creates the tailing off as the values of the steps get large:

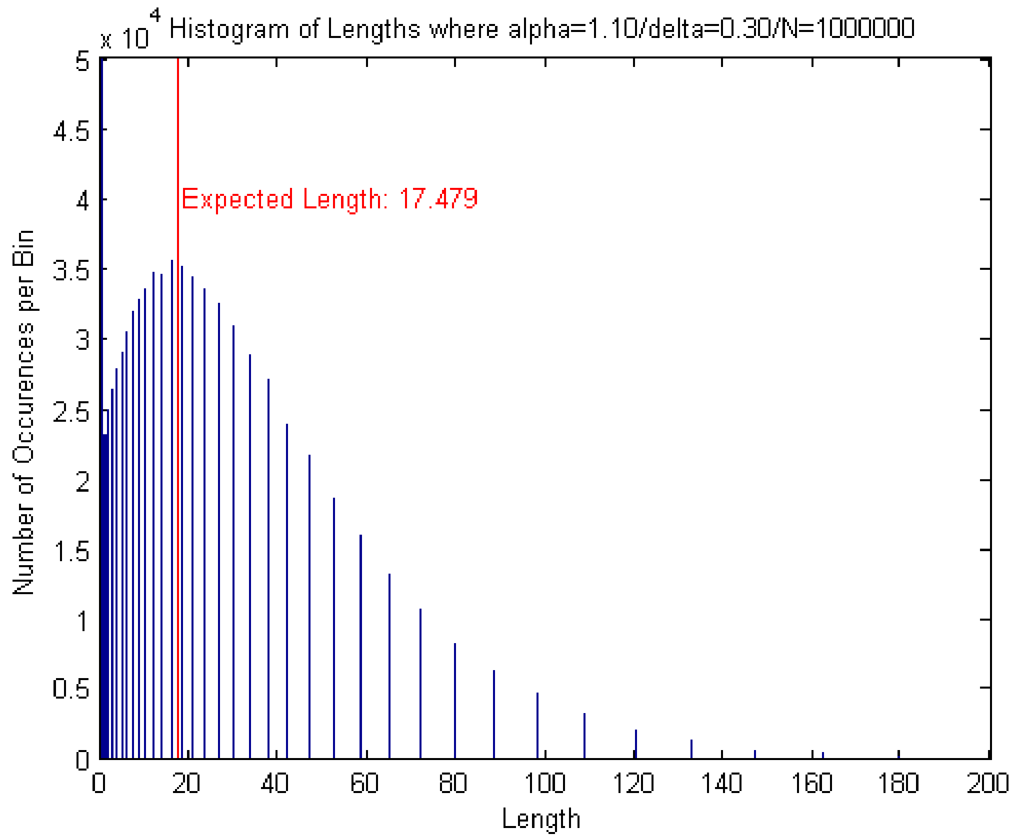

FIGURE 3. Histogram of Different Car Placements

Now, we need to find out what the optimal α and Δ should be for this exponential density function. This first simulation doesn't involve the sight parameter, because we will want to compare the two graphs later. In order to do this, we need to pick α and Δ values in a certain range in order to zero in on the correct pair. For each pair, we will run a large number of simulations to see what value gives the minimum expected length for finding the car. The graph below shows how this data becomes quite accurate when large sample sizes are taken. When only 1000 simulations are run, we see a very piecewise and choppy graph, but when the sample size is closer to one million, we get a nice graph that helps to verify our results. In the following graph, the different curves represent different α values and the x-axis denotes different Δ values. The expected lengths are shown on the y-axis:

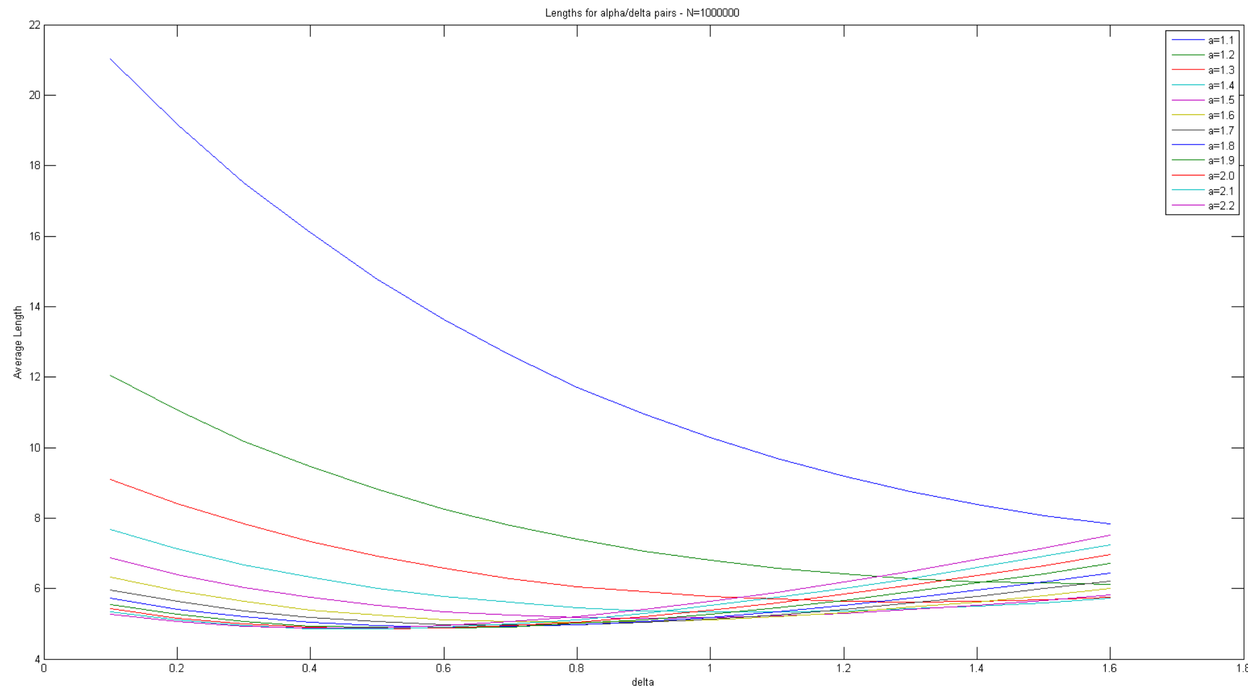

FIGURE 4. Optimization of Geometric Sequence without Sight Parameter

As we can see, the values tend to converge close to α = 2.1 and Δ = 0.5 with a length of 4.79. From here, we can choose smaller ranges around these values to get a more accurate result, but we are just using this graph to show how to get these optimal values, so we will not be taking this step here. This graph is a great representation of how the Monte Carlo Method can get us very good results when using a high sample size, and we can see that all of these curves appear to be differentiable. From here, we can use a similar method using two pivots instead of an α and Δ value in order to find the steps that need to be taken to minimize the length for an exponential distribution as opposed to a parabolic one.

In order to create a more realistic situation, we add in an additional sight parameter. This parameter is determined after each step using a normal distribution, as shown Figure 2. However, we decided to implement a condition such that the minimum sight value after any given step is equal to the mean sight parameter of the normal distribution. Therefore, at any point in the time, the sight is greater than or equal to the mean of the normal distribution. In Figure 2, therefore, the minimum sight is .2 and we name the sight parameter = .2. For each step, a random sight is chosen by the simulation in accordance with the normal distribution. If the car is spotted within this sight, the robot returns to the origin and the search is finished. If the car is not spotted, the robot returns to the origin and the search continues on in accordance with the geometric stepping pattern.

Now, we want to see what happens to the α and Δ values when we add in this new sight parameter and how the relationships between the variables change. The first graph below is a simulation that runs along a range of 10 α values that is one-tenth of a unit long. The resolution for the Δ values is the same. In this simulation, only one million data points were used to calculate the average length for each α/Δ pair. We see from this graph that the sight parameter makes everything a little messier as we now have two random variables instead of one:

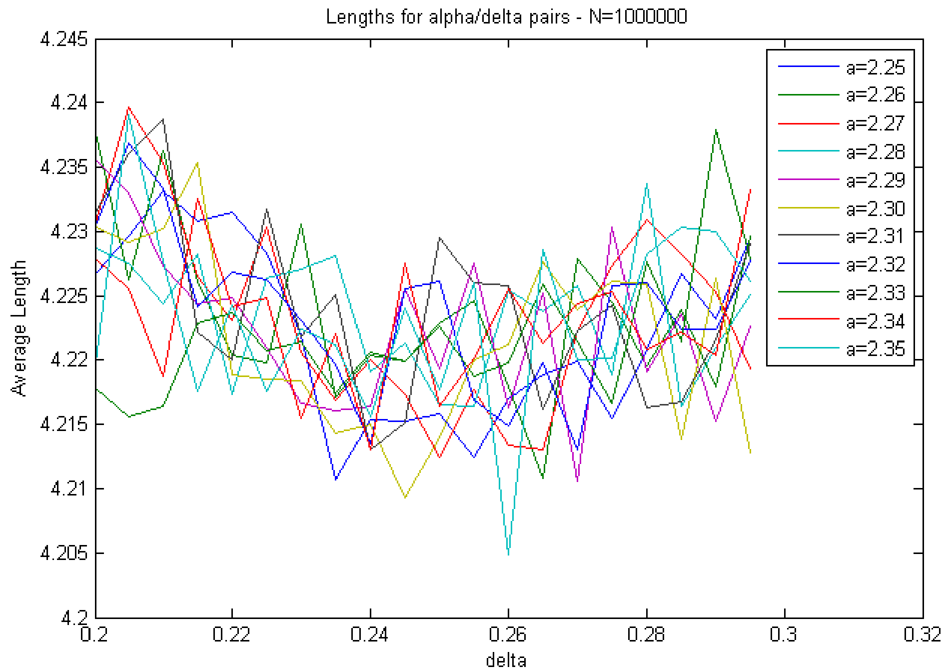

FIGURE 5. Optimization with Sight Parameter and N=1,000,000

Obviously, this graph is of little use to us as we cannot zero in on the correct α and Δ values. In order to fix this problem, we raised the number of iterations to one billion as we need quite a few orders of magnitude to get a good reading seeing that Monte Carlo simulations converge at a rate proportional to the square root of the number of iterations. The graph below is the same resolution as the previous graph with just one billion iterations instead of one million:

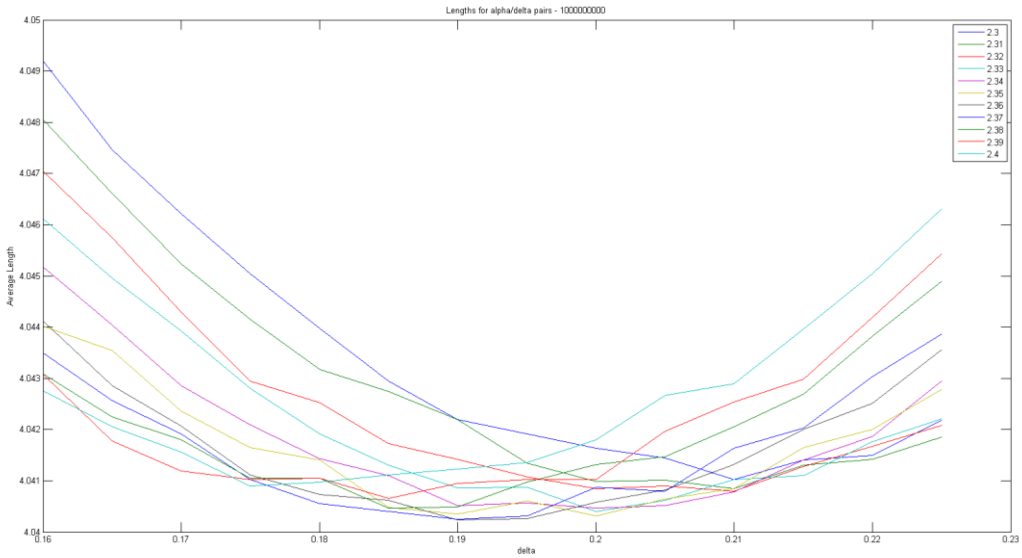

FIGURE 6. Optimization with Sight Parameter and N=1,000,000,000

Now we see the similar pattern of α lines showing up, and this is a graph that we can use to obtain the values we are looking for. In this graph, we see that α, Δ, and the minimum average L values are 2.36, 0.19, and 4.04 respectively.

Now that we have these values, we want to see what happens to the α and Δ values as the sight parameter is shifted up and down. We expect these values to change as the problem changes depending on how far one can see down the forest road. To find these relationships, we calculated the optimal α and Δ values for fifteen different sight values ranging from 0.10 to 0.25. The graphs showing the relationship are shown on the following pages:

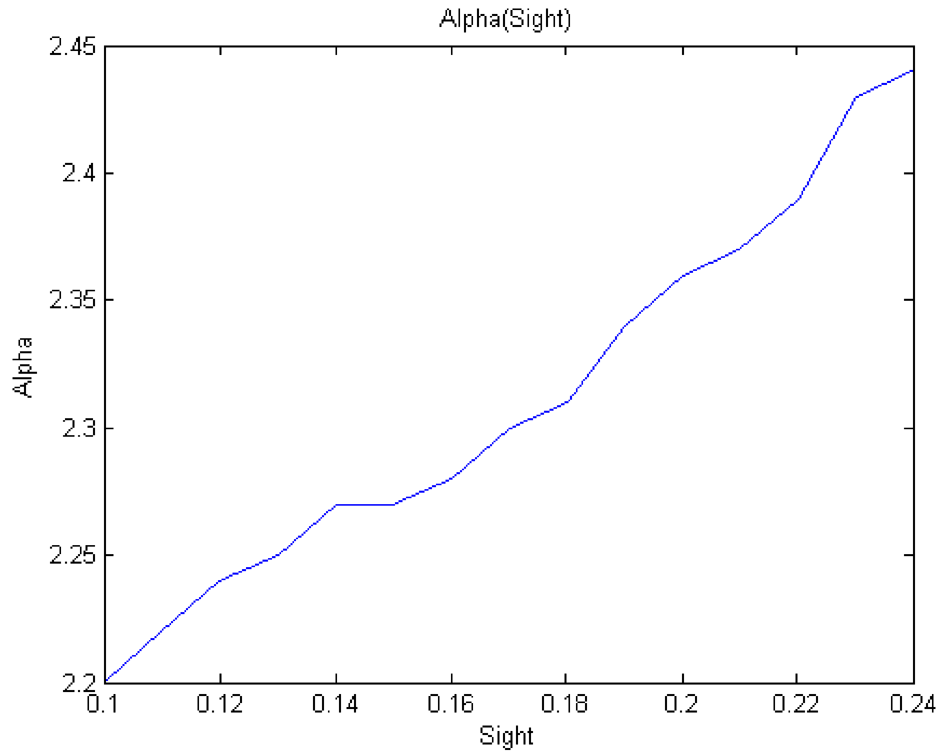

FIGURE 7. α vs. Sight

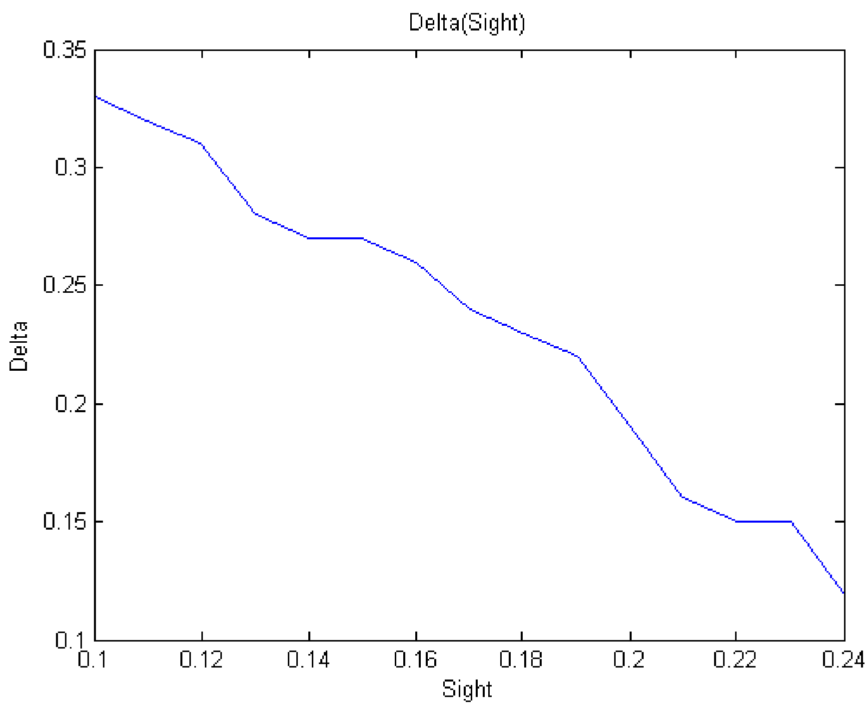

FIGURE 8. Δ vs. Sight

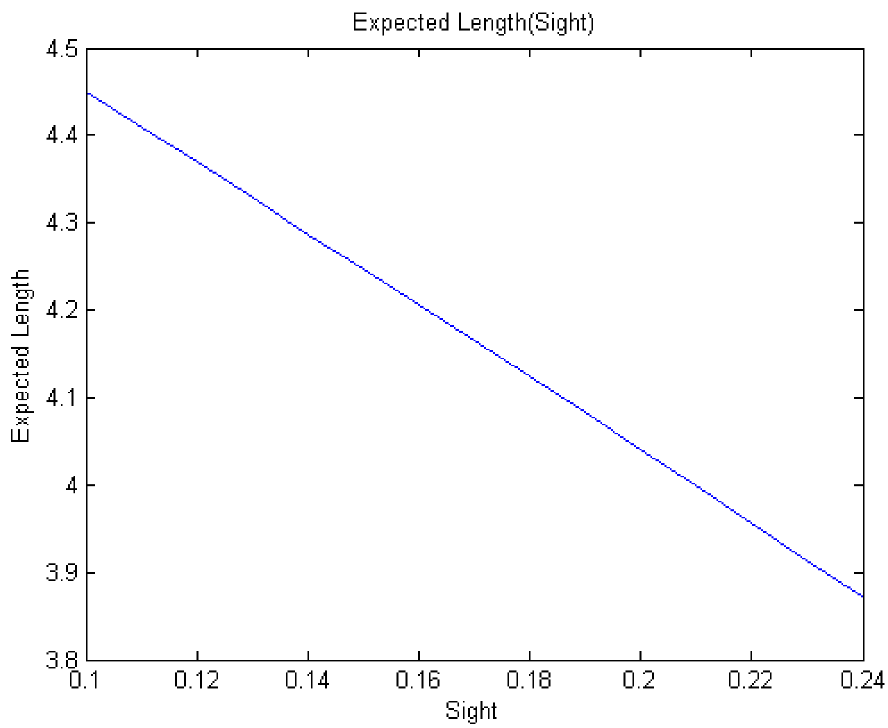

FIGURE 9. Average Length vs. Sight

From these figures, we see something interesting that applies to the physical world. In Fig. 6, we see that α grows proportional in a linear manner to the value of the sight parameter. This makes sense as α is the geometric ratio of the step values, so the further you can see, the longer each consequent step should be. Conversely, with the Δ values, we see that it is inversely proportional. This is because the more you can see down the road, the shorter your initial step needs to be. Finally, we see that the length gets shorter as you can see further down the road, and this just verifies that our model fulfills this relationship.

Conclusion

Our main goal was to determine the optimal parameters for sight and the geometric parameters α and Δ for a minimized search length. Being unable to do this analytically, we did Monte Carlo simulations to find the values numerically. First, we did preliminary simulations that did not involve the sight parameter. The preliminary results were 2.1, 0.5, and 4.79 for α, Δ, and L, respectively, which is represented in Figure 4.

After considering the parameter sight, we found those values to be 0.2, 2.36, 0.19, and 4.04 for sight, α, Δ, and L, respectively. Similar to Figure 4, Figure 6 represents the results with the parameter sight taken into consideration. In addition, we studied the relationships between sight and the other parameters. The result was the increase of α and decrease of Δ as the sight increases. Also, L decreases as the sight increases. Those relationships reflect what each parameter means in physical meaning. When you can see more, you can move a greater distance (α vs. sight) and reach your final destination faster (L vs. sight) as well as starting with a smaller initial step (Δ vs. sight). On the other hand, the degree of dependence in those relationships cannot be determined exactly since only 15 points were used to visualize the dependence graphically. While the graphs for (α vs. sight) and (Δ vs. sight) were not very smooth, the graph displaying (L vs. sight) was quite smooth and suggests an inverse linear relationship. Those relationships can be seen in Figures 7 through 9.

Reference

1. Yu Baryshnikov and V Zharnitsky,
Search on the brink of chaos, Nonlinearity, 25 (2012), 3023–3047,
doi:10.1088/0951-7715/25/11/3023

2. Vincent Dumoulin and Felix Thouin.
A Ballistic Monte Carlo Approximation of π, physics.pop-ph, (2014).