Evaluating Levy Flight Parameters for Random Searches in a 2D Space

by

Mukul Kumar Singh

Submitted to the
Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

February 2013

© 2013 Massachusetts Institute of Technology. All rights reserved.

Signature of Author: ______________________________
Department of Mechanical Engineering
January 31, 2013

Certified by: ______________________________
James H. Williams, Jr.
Professor of Applied Mechanics
Thesis Supervisor

Accepted by: ______________________________
Anette Hosoi
Associate Professor of Mechanical Engineering
Undergraduate Officer

Evaluating Lévy Flight Parameters for Random Searches in a 2D Space

by

Mukul Kumar Singh

Submitted to the Department of Mechanical Engineering
on January 31, 2013 in Partial Fulfillment of the
Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

Abstract

It is experimentally known that the flight lengths of random searches by foragers such as honey bees statistically belong to a power law distribution[1]. Optimality of such random searches has been a topic of extensive research because knowing their optimal parameters may help applied sciences. Viswanathan et al.[2] have shown the inverse-square power law to be the optimal law for such random searches. This thesis explores the capability of the model presented in [2] such that it can be applied to Unmanned Autonomous Vehicles (UAVs). The thesis also identifies the *minimum flight length*, l_{min}, as an important factor that needs to be controlled based on the UAV's sensor range. We present a theoretical l_{min} as an explicit function of the sensor range, r_v, and an estimated target density, ρ.

Thesis Supervisor: James H. Williams, Jr.
Title: Professor of Applied Mechanics

Acknowledgment

I want to thank Professor Williams for his most appreciated patience in working with me. I want to thank him for guiding me through this process, giving me enough freedom so I could take my own directions, for the valuable lessons, and help me with my the formatting. It would not have simply been possible to achieve the shape of this thesis without the presence of Professor William's extraordinary personality. I am grateful for the financial contribution of the DDG-1000 Program Manager (NAVSEA PMS 500) and the technical guidance of the DDG-1000 Ship Design Manager (NAVSEA 05D).

Furthermore, I would like to take this section as a written opportunity to thank some individuals from whom I have had an incredible support in last five years.

First, I can't thank enough Dean Henderson. Had it not been his continuous support for last four years, I simply wouldn't have even been able to reach at the state of even initiating this thesis. Dean Henderson's support and encouragement have been uncanny from the day I stepped into MIT from my town in India. I want to thank him for he, alone, made me believe that my degree from MIT is possible.

I want to thank Brandy Baker, who like Dean Henderson, made me truly feel Department of Mechanical Engineering home. I want to thank her for her time and really pushing me through a lot of things.

I want to thank MIT Financial Aid Office, who has been very supportive all these years. I want to especially thank Claudia Batte, Jason Marsala, and Dwayne Daughtry for making my worries their own. Danielle Guichard-Ashbrook, and Maria Brennan from International Student Office for always being very helpful.

Friends and colleagues who I have interacted during my time at MIT, whose smiles, discussions, and jokes helped me feel the belongingness here, and helped me get through MIT. I want to especially thank a great individual and fortunately my friend Abiy Tassisa, MIT'12, currently pursuing Masters of Science at MIT. Abiy came to MIT from Ethiopia as an International Student. Abiy has, numerous times, selflessly made efforts to help me not just in academics, but also through hard times at MIT as an International student. And with no exception, Abiy, once again, has helped me through formatting, checking my assumptions, giving organized comments, and a free form discussions over several meetings that resulted into a major section of my thesis. I want to also thank my friend Raghu Mahajan, MIT'11, Owusu Ansah Agyeman Badu, MIT'11, and Prosper Nyo, MIT'12 who offered their help.

I want to give a special thanks to Dr. Harrison H. Chin, Professor Pierre Lermusiaux, Professor John Leonard, and Professor Lallit Anand for their time and academic advise. Maryanne Kirkbride, and my ESL instructor Patricia Brennecke for their support for past four years. I want to thank Dr. Chintan Vaishnav at MIT for his invaluable spiritual and philosophical guidance.

I want to thank my parents, and my mother especially, as her only child, the sacrifices she has made to see me graduate from here is unparallel.

Contents

List of Figures

1 Introduction

A recent observation made about foraging animals is that foragers when in no or limited prior knowledge of food show searching patterns that have special characteristics. The patterns are different from what can be seen in Brownian motion, the random walk by particles diffusing in a liquid. Foragers sometimes take long paths in just one direction. This strategy is found to be the key to the foragers' success in finding food rapidly in an unknown environment [1].

It is shown experimentally that these foragers often take an optimal random searching strategy [1]. The experimental method in [1] includes tracking honeybees using harmonic radar tracking, and fitting the found flight lengths to a probability distribution. The flight lengths fit in to a Lévy like characteristic with the optimal parameters proposed earlier and independently in [2]. Viswanathan et al. analytically show that the optimal strategy for random searches is an inverse-square power law distribution for any general case based on **an idealized foraging model**. Their analytical model closely represents the forager-food relationship.

We take Viswanathan et al.'s analytical results and reapply them to a controllable application such as Unmanned Autonomous Vehicles (UAVs). UAVs, unlike natural objects, will require a precise probability distribution so that it can generate random walks that statistically fit in a desired distribution.

1.1 The Power-Law Distribution

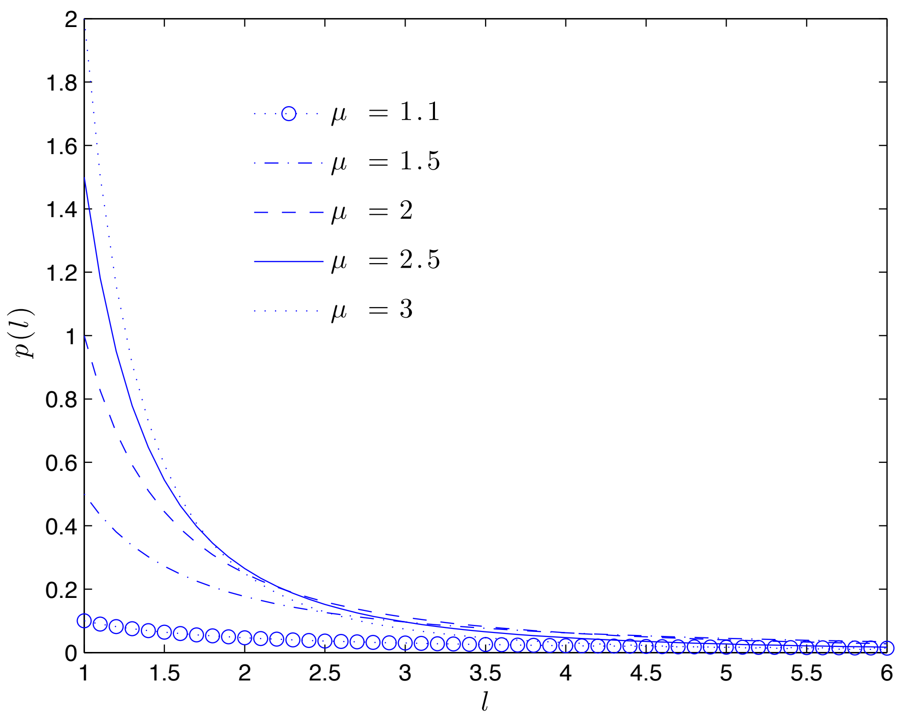
Power-law distributions have a wide range of appearances in different fields. In a target searching context, flight lengths, l, are said to have the **power-law distribution** when all these lengths are drawn from a probability distribution of the form

$$p(l) \propto l^{-\mu}, \tag{1.1}$$

where μ is the *scaling parameter*.

A more general **Lévy distribution** can be classified as a power-law distribution where the scaling parameter, μ, lies in the range $1 < \mu \leq 3$. Scaling parameter $\mu \leq 1$ can not be normalized, and power-law distribution when $\mu \geq 3$ turns into a Gaussian distribution [4]. Thus, a Lévy flight consists of flight lengths that obey the power-law distribution with $1 < \mu \leq 3$.

It is clear from the Relation (1.1) that when l approaches zero, the probability density $p(l)$ diverges which suggests that l must have a lower bound, l_{min}. Figure (1) shows probability distributions for a few arbitrary values of μ. We rewrite (1.1) in an equation form:



ss

Figure 1: $p(l)$ for different values of μ.

$$p(l) = C\, l^{-\mu}, \tag{1.2}$$

where C is the normalization constant calculated as follows:
With a lower bound specified, probability density $p(l)$ satisfies

$$\int_{l_{min}}^{\infty} p(l)\; dl = \int_{l_{min}}^{\infty} C\, l^{-\mu} dl = 1. \tag{1.3}$$

For the continuous probability case, C can be easily calculated from Eq.(1.3). For $\mu > 1$, the normalization constant can be calculated and gives Relation (1.2) the following form for the **continuous case**:

$$p(l) = \frac{\mu - 1}{l_{min}} \left(\frac{l}{l_{min}}\right)^{-\mu}. \tag{1.4}$$

Similarly, for the **discrete case**, C is calculated from $\sum_{l=l_{min}}^{\infty} C\, p(l) = 1$. In

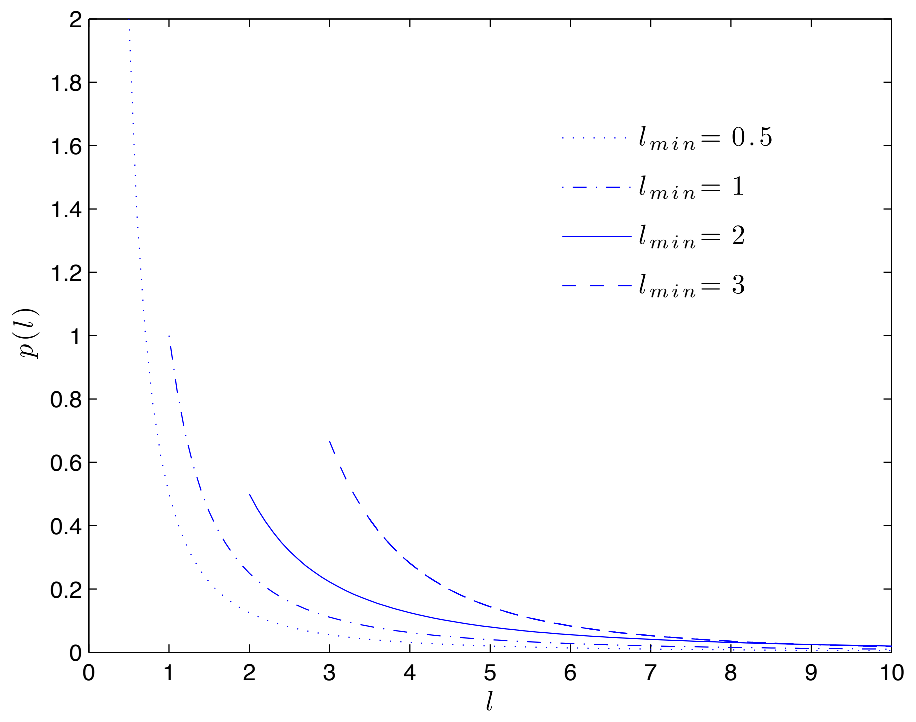


Figure 2: $p(l)$ for different values of l_{min} at $\mu = 2$.

discrete case, C takes the following form:

$$C = \left(\sum_{n=0}^{\infty} (n + l_{min})^{-\mu} \right)^{-1}. \tag{1.5}$$

1.2 The Idealized Foraging Model

Assuming a **two dimensional** random search, the analytical model studied by Viswanathan et al. is summarized in following points. The forager follows these rules:

1. The forager moves from one point to another until it locates a target.

2. In the **two dimensional case**, the *angle* that the forager takes between one flight to the next flight is drawn from a **uniform distribution** of angles in a range of $[0, 2\pi]$.

3. If the forager locates a target in its vicinity, the forager moves to the

target in a straight line, i.e., the closest distance between the forager and the target.

4. The forager resumes its search according to rules (1) and (2) as soon as it can again no longer locate a target, i.e., its flight lengths, once again, are drawn from the same probability distribution as rule (1).

One *flight length* can be defined as the distance traveled by the forager between one point to another without stopping and changing the angle of its path. The forager may take several of such flight lengths to find one target.

Using the analytical method in [2], the mean flight length is derived to be

$$l_{mean} = \left(\frac{\mu - 1}{2 - \mu}\right)\left(\frac{\lambda^{2-\mu} - r_v^{2-\mu}}{r_v^{1-\mu}}\right) + \frac{\lambda^{2-\mu}}{r_v^{1-\mu}}, \tag{1.6}$$

where μ is the scaling parameter of the power-law distribution, r_v is the radius of the circle which represents forager's vicinity, and λ is the mean free path between the successive targets. Figure (3) shows the variation of analytical l_{mean} value for an increasing μ for different target area densities.

The search efficiency η, a function of μ, is defined in terms of mean flight length and the number of flights taken to find a target. Figure (4) shows η times λ as it varies with changing μ.

$$\eta(\mu) = \frac{1}{l_{mean}\ N} \tag{1.7}$$

2 The Model and Optimized Parameter

2.1 The Modified UAV Model

We assume that the targets are randomly distributed and are stationary. The modified UAV model does not follow rule (3) in the idealized forager model discussed earlier. Since, in many engineering applications, most of the random searches done are intended to only *locate* an object of interest, we are only concerned of locating the target as fast as possible. Once the object is located, often an operator decides what to do with the found target. For instance, in a case of a rescue mission, finding drowning people is the sole aim of a UAV. It cannot further spend its time in interacting with the just found target, it must move on to its search for finding other targets. In such a case, however, there can be another system that can be released from the

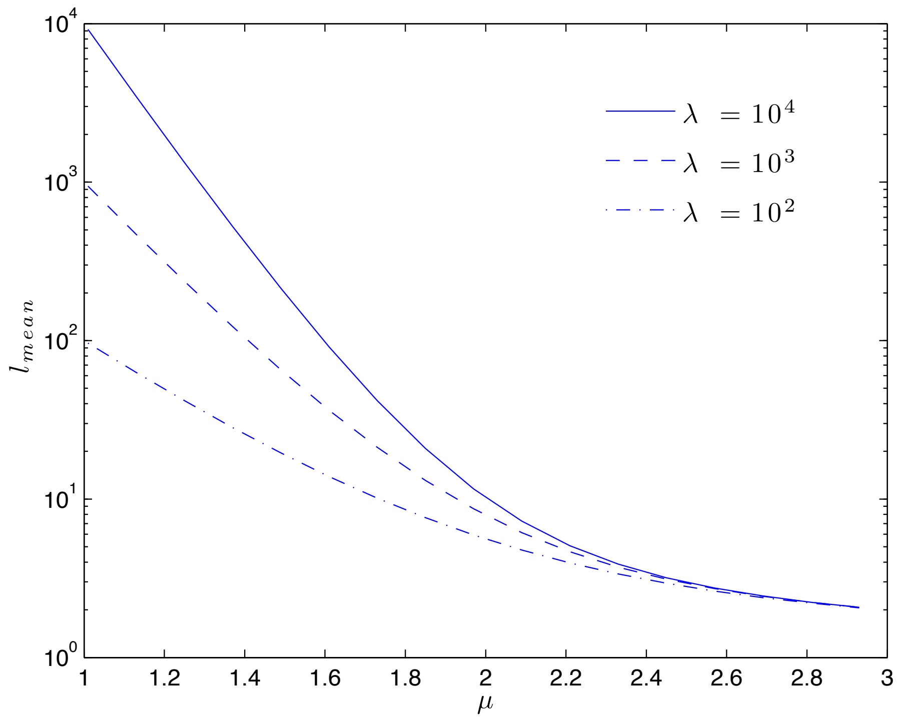


Figure 3: l_{mean} versus μ for $\lambda = 10^4, 10^3$, and 10^2.

searching UAV to send some relief items to the person until the found person waits for the complete help.

This modified model can be summarized similarly to the forager model by excluding the rule (2) as follows:

1. The Unmanned Autonomous Vehicle (UAV) moves from one point to another until it *locates* a target, i.e UAV keeps drawing lengths to follow from a predetermined probability distribution. The probability distribution in our case will be a power-law distribution with a fixed μ.

2. If the UAV locates a target in its vicinity, the sensor range of radius r_v, the UAV sends out the information to an operator outside.

3. Without stopping, the UAV moves on to its search i.e., the next flight length is drawn from the power-law distribution with a pre-determined value of scale factor μ. The angle UAV takes is drawn from a uniformly distributed set of range $[0, 2\pi]$.

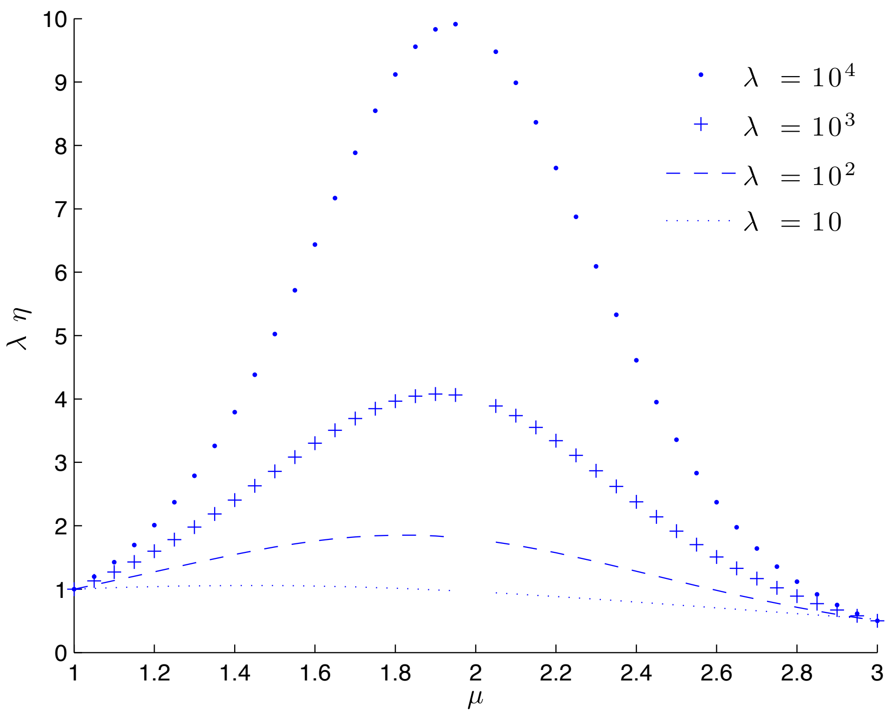


Figure 4: $\lambda\eta$ plotted for different values of μ and λ.

4. The mapping of the region has not been specified - the UAV can revisit any region.

2.2 Search Efficiency and Optimal Scaling Parameter

Let's take λ to be the *mean free path* between the successive *targets*, l_{mean} to be the mean UAV flight length over the N steps traveled by the UAV, r_v the range of the sensor mounted, and ρ to be the target area density. Viswanathan et al. defines the efficiency η of one *complete path* of discovering the target to be the ratio of the number of target discovered over the total distance traveled.

If l_{mean} is the mean flight length over N steps taken by the UAV to discover its first target, the total distance traveled by the UAV can be equated to N times l_{mean}, and the efficiency η is given by

$$\eta = \frac{1}{N\, l_{mean}} \tag{2.1}$$

The Eq.(2.1) is the efficiency of one such path, i.e., the path from one target to finding the next, where λ is the mean free path between the two successive target sites.

It is important to note while reproducing Figure (4) that N in Eq.(2.1) is variable which changes with varying μ. Viswanathan et al. presents the variable N for two cases: the forager finishes eating the target (destructive foraging), and non-destructive foraging. UAV searches, can have similar divisions: once the target is located it could either be moved away or remain in the search area. Figure (4) is produced taking N to be for the non-destructive case:

$$N = \left(\frac{\lambda}{r_v}\right)^{(\mu-1)/2} \tag{2.2}$$

3 Generating Lévy Flights

3.1 Simulating Random Searches with Arbitrary Parametric Values

Simulating random searches includes generating Lévy flights drawn from the power-law distribution as described in the Introduction section, and angles drawn from a uniform distribution. This section explores a basic methodology of generating these flights for different values of scaling parameter μ.

The method used to generate the power-law distributed flight lengths is known as the **transformation method**. The method is one of the simplest and elegant methods for producing such probability distribution for generating both, **discrete** and **continuous** distributions [4]. It maps a uniformly distributed data to the desired distribution in the following manner:

The method is based on first calculating the cumulative density function of our power-law density function, and then, inverting the cumulative density function. The CDF takes uniformly distributed points as input to generate the desired set of Lévy distributed flight lengths.

Cumulative density function (CDF) when continuous

$$\begin{aligned} CDF &= \int_{l_{min}}^{\infty} p(l)\,\mathrm{d}l \\ &= \int_{l_{min}}^{\infty} C\,l^{-\mu}\,\mathrm{d}l \quad \text{where } C = (\mu-1)\,l_{min}^{-\mu} \\ CDF &= 1 - \left(\frac{l}{l_{min}}\right)^{1-\mu} \end{aligned} \tag{3.1}$$

Inverting the CDF function and the transformation

$$\begin{aligned} l_j &= l_{min}\ (1 - CDF)^{\frac{1}{-\mu+1}} \\ l_j &= l_{min}\ (1 - u_j)^{\frac{1}{-\mu+1}} \\ \mathbf{l} &= l_{min}\ (1 - \mathbf{u})^{\frac{1}{-\mu+1}} \end{aligned} \tag{3.2}$$

Eq.(3.2) shows the transformation method - $\mathbf{u}$ as input, is uniformly distributed set in range $[0, 1)$, that is transformed in to the output $\mathbf{l}$, the flight lengths between $[l_{min}, \infty)$ obeying power-law distribution of scaling parameter μ. In other words, if u_1 and u_2 belong to a uniform distribution in $[0, 1)$, plugging these value in Eq.(3.2) will result in two points, say, l_1 and l_2 that belong to the power-law distribution of parameter μ. The MATLAB code that implements this method can be found in Appendix A.

3.2 Validating Power-Law Distribution

To check whether the flight lengths $\mathbf{l}$ generated in Section (3.1) do belong to the desired power-law distribution, we *estimate* the value of μ for the generated flight lengths. The method used here is referred as the *method of maximum likelihood* (MLE) [4]. The maximum likelihood estimator (MLE) for continuous power-law distribution is

$$\mu_{est} = 1 + n \left[\sum_{j=1}^{n} \ln \frac{l_j}{l_{min}} \right]^{-1}. \tag{3.3}$$

Figure (5) shows the estimated scaling parameter, μ_{est}, for l_j that are being generated using Eq.(3.3) for $\mu = 2$. We observe that μ_{est} quickly catches up to a close value of actual μ that is been used to generate these sample flight lengths, and thus validates our method described in Section (3.1). The MATLAB code in Appendix A has been provided that produces this result.

4 Estimating the Minimum Flight Length: l_{min}

In a previous section, we discussed that optimal random searches are achieved for $\mu \approx 2$. In the case of UAV search, the robot at any given location and instant of time needs an instruction that tells it a direction it should take and how long it should travel. Our attempt in this section is to identify a parameter that will ultimately be used in a code for a practical UAV search.

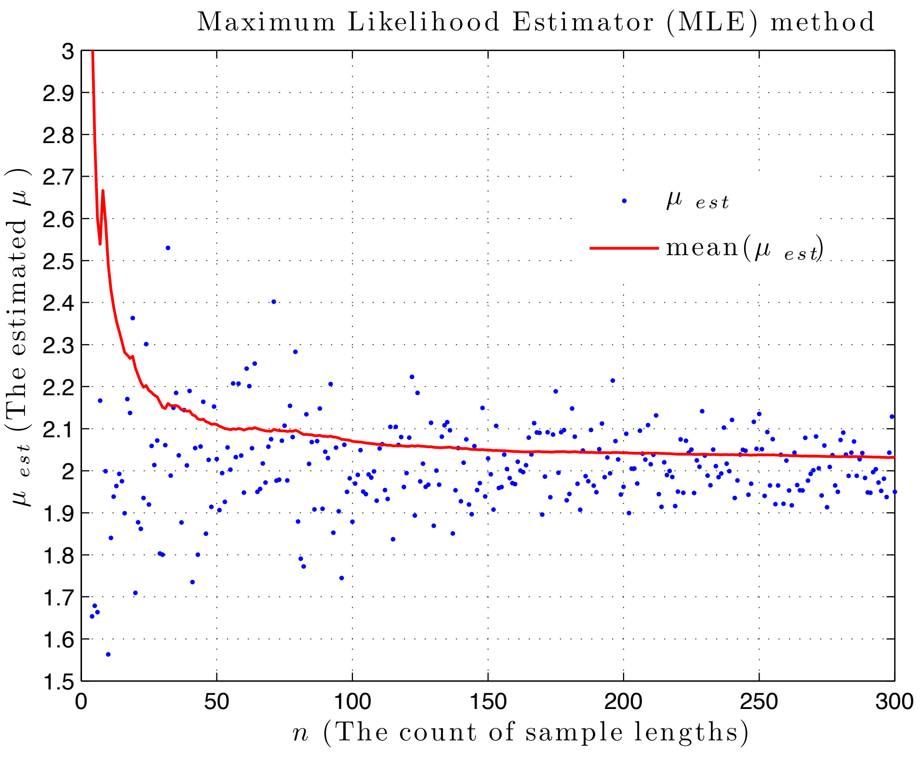


Figure 5: μ_{est} for flights generated from Eq.(3.2) for $\mu = 2$.

What we consider here is an inverse of the problem of foraging. In the problem of foraging, we collect the flight lengths animals take and see what probability distribution they fit. Now we are interested to instruct a robot for a given probability distribution.

We identify l_{min} to be an important parameter for the following reason. In a UAV search context, the flight lengths drawn with a higher l_{min} will have a higher mean flight length than the mean flight length with a lower value of l_{min}. Figure (9) validates this point. It also shows l_{mean} is highly sensitive to l_{min} when the scaling parameter, μ, is a close value to 2.

Motivation: An intuitive way of looking at it is to think about having a very small minimum flight length compared with the UAV sensor's range. Since it's a low value of l_{min}, from Figure (2), it will lie in far left region on the x axis. Since the power-law distribution is normalized, the values closer to l_{min} have a higher probability than the other *larger* flight lengths which fall further on the right side of the axis. A CDF has been plotted to show this behavior.

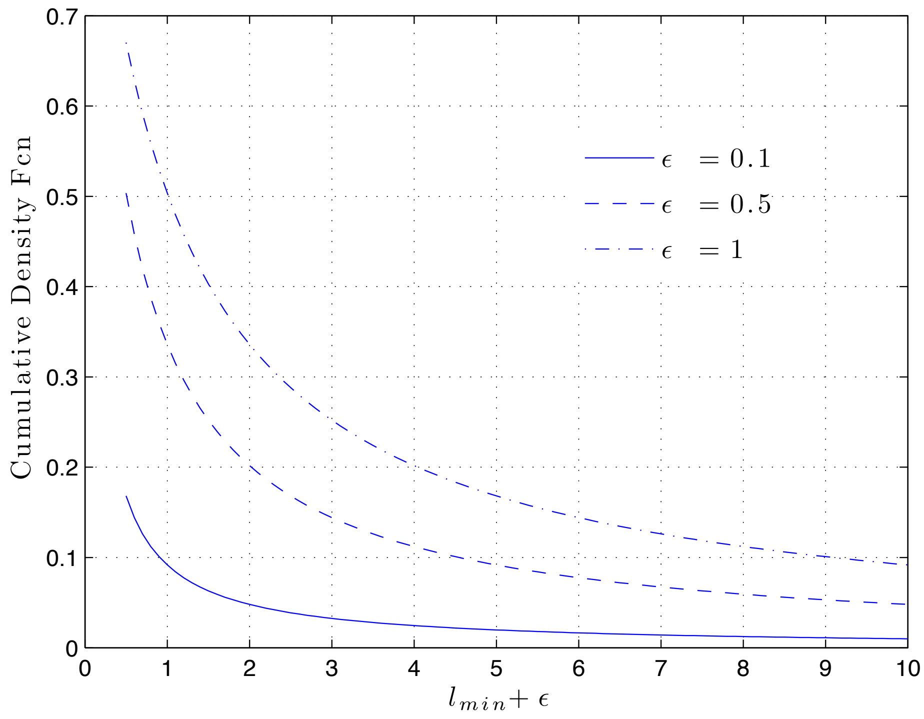


Figure 6: CDF between a varying l_{min}, and $l_{min} + \epsilon$, where ϵ is a small value.

In Figure (6), we vary l_{min} to see its effect on the probability distribution in its nearby region. ϵ parametrizes this small region. The region, l_{min} with a fixed value of ϵ, have a higher cumulative probability distribution than from the region with the bigger l_{min} values. Thus, when l_{min} is low, the probability of the sample lengths that are a close value of l_{min} is high, and vice versa. For instance, let's say when we draw twenty flight lengths from one of these distributions, we find fifteen out of these twenty flight lengths to have a close value to l_{min}, which is a very low value compared with the sensor range, then the UAV would barely cover any region with these samples. From a similar scenario, where l_{min} is very large compared with r_v, then that would leave out a lot of area uncovered. Thus, one can see how inefficient these flight lengths will be for our search, and the need to determine l_{min} carefully. Figure (7) is a schematic presentation of this observation. Furthermore, one can see from Figure (8) that l_{mean} stays positive for only a range of r_v. Therefore, it is important to identify the constraints on l_{min}.

Methodology: Analytical solution for mean flight length Eq.(4.1) provided by Viswanathan et al. after simplifying for λ gives us a co-relation between l_{mean} and r_v. Assuming that - the value of μ has been decided to be taken a close number to 2 to keep the random search optimal, ρ is *roughly known* to the operator based on operator's knowledge of the search field, r_v, the sensor range, is known (as it is reasonable to assume most sensor are shipped with specifications) - we compute l_{mean} using Eq.(4.1). Now to obtain a corresponding value of l_{min} for input values of ρ, μ, and r_v, we calculate the mean flight length from the probability distribution, Eq.(4.5), we thus eliminate l_{mean} from these equations and obtain the desired value of l_{min} in terms of ρ, μ, and r_v.

Detailed Derivation: The analytical model given by Viswanathan et al. that correlates *mean* flight length to μ, the scaling parameter, r_v, the range of the forager, and λ, the mean free path between the successive target sites, is as follows:

$$l_{mean} = \left(\frac{\mu - 1}{2 - \mu}\right)\left(\frac{\lambda^{2-\mu} - r_v^{2-\mu}}{r_v^{1-\mu}}\right) + \frac{\lambda^{2-\mu}}{r_v^{1-\mu}} \tag{4.1}$$

For the two dimension case, mean free path between successive targets, λ, and target area density, ρ are correlated as follows [2]:

$$\lambda = (2\rho r_v)^{-1} \tag{4.2}$$

Plugging (4.2) into (4.1) takes the following form that correlates r_v with

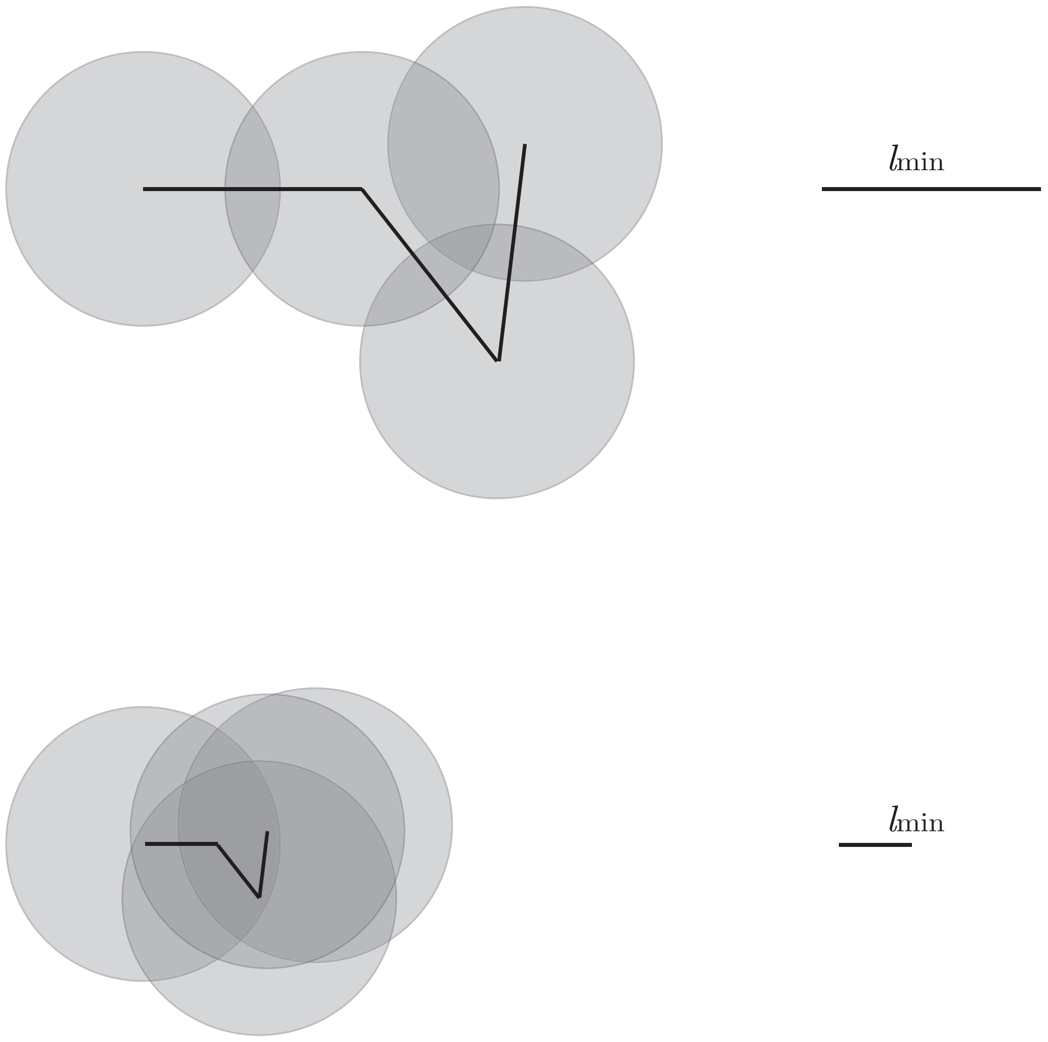


Figure 7: Schematic presentation of the effect of – a comparable l_{min} and a low value of l_{min} compared with the sensor range r_v – on search efficiency. The demo shows the three flight lengths which are minimum flight lengths l_{min}.

l_{mean} :

$$l_{mean} = \big((-k1 + 1)\, k2\big)\, r_v^{2\mu-3} + k1\ r_v \tag{4.3}$$

where $k1 = \left(\dfrac{\mu - 1}{\mu - 2}\right)$ and $k2 = (2\rho)^{\mu-2}$.

It can be noted from Eq.(4.1) that for an exact $\mu = 2$, the value of l_{mean} becomes infinity. To inspect the behavior of Eq.(4.3), a plot between l_{mean} and r_v has been shown in Figure (8). Here, to keep l_{mean} value finite, we arbitrarily choose $\mu = 2.01$ considering it to be a close value to 2, and $\rho = 1$.

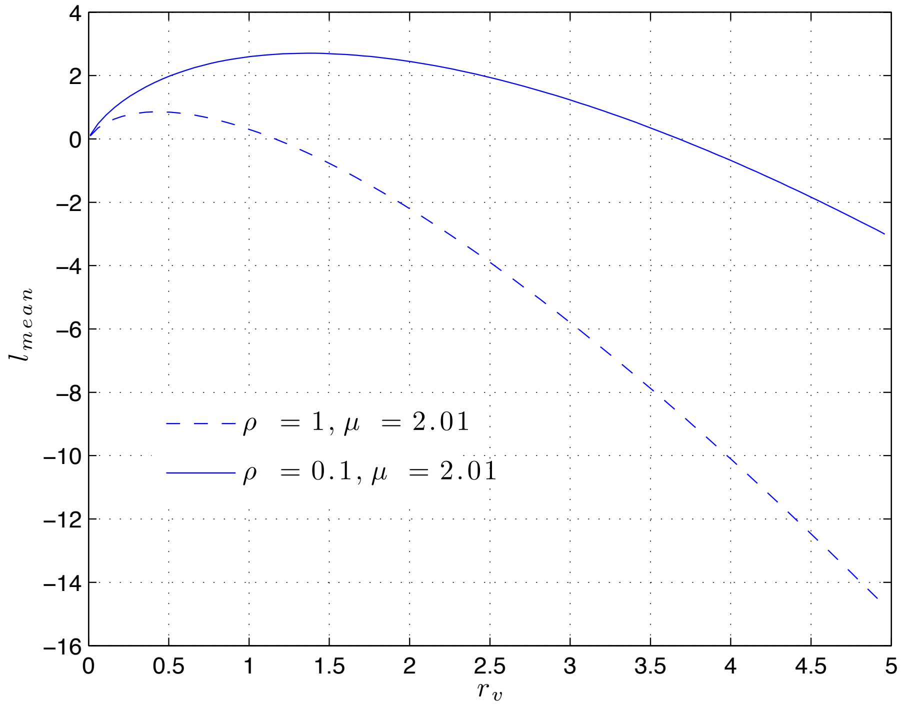


Figure 8: l_{mean} and r_v for $\mu = 2.01$ and different target site densities, ρ.

The plot shows an interesting result as we see that l_{mean} is only valid, i.e positive, of a certain range of r_v. Further exploring the co-relation (4.3), we observe the target density is an important factor in keeping the l_{mean} valid for a larger value of r_v.

Continuous power-law distribution

$$p(l) = \left(\frac{\mu - 1}{l_{min}}\right)\left(\frac{l}{l_{min}}\right)^{-\mu} \tag{4.4}$$

By definition, the *mean* of a given probability density function $p(x)$ is calculated as follows :

$$x_{mean} = \int_a^b x\ p(x)\ dx \tag{4.5}$$

Eq.(4.5) for the power-law distribution can be written as

$$\begin{aligned} l_{mean} &= \int\limits_{l_{min}}^{\infty} l \left(\frac{\mu-1}{l_{min}}\right)\left(\frac{l}{l_{min}}\right)^{-\mu} dl \qquad (4.6) \\ &= \int\limits_{l_{min}}^{\infty} \frac{(\mu-1)\ l^{-\mu+1}}{l_{min}^{-\mu+1}} dl \\ &= \left(\frac{\mu-1}{l_{min}^{-\mu+1}}\right)\left[\frac{l^{-\mu+2}}{-\mu+2}\right]_{l_{min}}^{\infty} \\ &= \left(\frac{\mu-1}{l_{min}^{-\mu+1}}\right)\lim_{b\to\infty}\left[\frac{l^{-\mu+2}}{-\mu+2}\right]_{l_{min}}^{b} \\ &= \left(\frac{\mu-1}{l_{min}^{-\mu+1}}\right)\lim_{b\to\infty}\left[\frac{b^{-\mu+2}}{-\mu+2} - \frac{l_{min}^{-\mu+2}}{-\mu+2}\right] \end{aligned}$$

$$l_{mean} = l_{min}^{\mu-1}\left(\frac{\mu-1}{\mu-2}\right)\lim_{b\to\infty}\left[l_{min}^{2-\mu} - b^{2-\mu}\right] \qquad (4.7)$$

The second term in Eq.(4.7) diverges when $\mu \leq 2$; for $\mu > 2$, the term $b^{2-\mu}$ converges and l_{mean} can be written as

$$l_{mean} = \left(\frac{\mu-1}{\mu-2}\right) l_{min} \quad \text{for } \mu > 2 \qquad (4.8)$$

Eq.(4.8) can be written in a desirable state for calculating l_{min}

$$l_{min} = l_{mean}\left(\frac{\mu-2}{\mu-1}\right) \quad \text{for } \mu > 2 \qquad (4.9)$$

Eq.(4.9) when plugged in for l_{mean} results into a direct correlation relating minimum flight length l_{min} to sensor's range r_v and target area density λ.

$$\boxed{l_{min} = -(2\rho)^{\mu-2}\left(1+\frac{\mu-2}{\mu-1}\right) r_v^{2\mu-1} + r_v \quad \text{for } \mu > 2} \qquad (4.10)$$

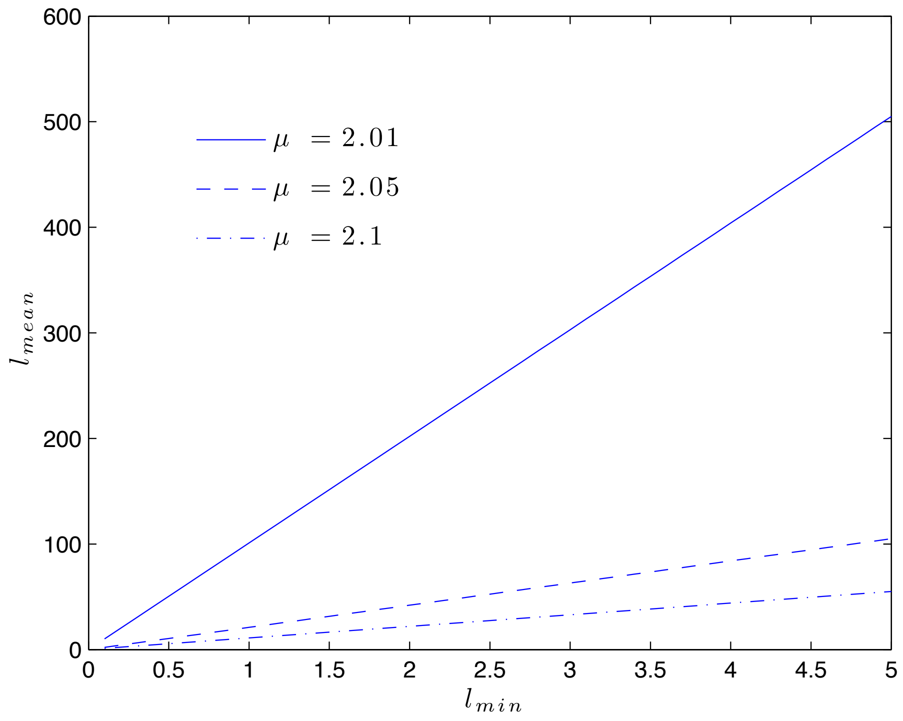


Figure 9: l_{mean} as it varies with changing l_{min} - Eq.(4.8).

5 Conclusion

As mentioned in Introduction section, the analytical model has been used extensively in the animal foraging research community, but the aim of this paper is to extend the model to a more controlled application such as UAVs. The thesis focuses on exploring the required technical aspects of transferring a random search that is been classically proposed by Viswanathan et al. to an Unmanned Autonomous Vehicle. The thesis discusses the analytical aspects of the model and random search parameters, but it doesn't include a precise implementation of a specific sensor for UAV. We assume that UAVs and forager have the similar searching mechanism for random searches. Using Viswanathan et al. 's methods, we take inverse square law (*scaling parameter*: $\mu = 2$) to be the most optimal value for random searches as shown in Figure (4) in Section (2.2).

The report talks extensively about the importance of the minimum flight length and its relation to the sensor range as depicted in schematic 7. The

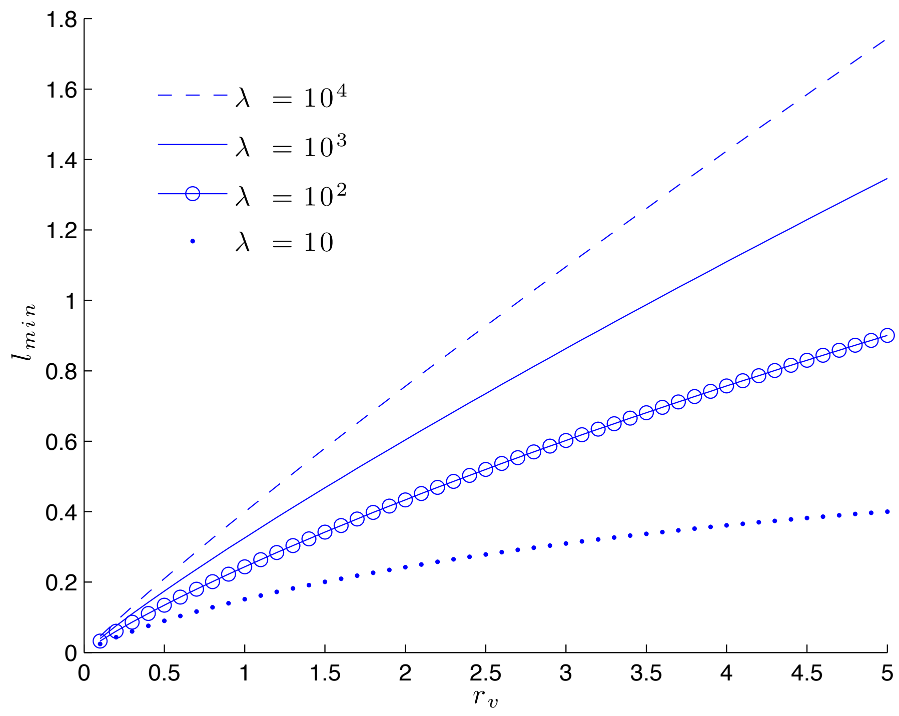


Figure 10: Estimating l_{min} for different r_v and λ at $\mu = 2.05$ using Equation (4.10).

Eq.(4.10) finds the minimum flight length, l_{min}, to be a function of UAV sensor's range (r_v) and a roughly known target-site density (ρ) or mean free path between successive target (λ), where ρ and λ are interrelated by Eq.(4.2). Figure (10) shows a direct relationship between the sensor range and minimum flight length. As one can predict from Schematic (7), the minimum flight length l_{min} should increase as the range of the sensor r_v increases as to keep the search optimal. We obtain a confirming result for varying target density. This relation, Eq.(4.10), derived from Viswanathan et al. 's analytical solution helps by optimally deciding minimum flight length for UAVs with various ranges for different target densities.

References

[1] Reynolds, Andrew M., Smith, Alan D., Reynolds, Don R., Carreck Norman L., and Osborne, Juliet L., “Honeybees perform optimal scale-free searching flights when attempting to locate a food source.” Journal of Experimental Biology, 210: 3763-3770. 2007.

[2] Viswanathan, G. M., Buldyrev, Sergey V., Havlin, S., da Luz, M. G. E., Raposo, E. P. and Stanley, H. E.,“Optimizing the success of random searches.” Nature 401, 911-914. 1999.

[3] Bouchaud, J.-P. and Georges, “A. Anomalous diffusion in disordered media: statistical mechanisms, models and physical applications.” Phys. Rep. 195, 127-293. 1990.

[4] Clauset, A., Rohilla Shalizi, C., and Newman, M. E. J. “Power-law distributions in empirical data.” SIAM Review 51, 661-703. 2009.

[5] Lenagh, W., Dasgupta, P., “Lévy Distributed Search Behaviors for Mobile Target Locating and Tracking.” Proceedings of the 19th Conference on Behavior Representation in Modeling and Simulation, Charleston, SC, 21 - 24 March 2010.

A MATLAB Code

```
%% plotting 'Normalized' mu =2 for different values of lmin
mu = 3;
lmin = 3%  1, 2, 3
l=[3:0.1:10]; % 0.5, 1, 2, 3
C=(mu-1)*lmin.^(mu-1);
p=C*l.^(-mu);
plot(l,p,'--')
hold on
% Create xlabel
xlabel({'$\l$'},'Interpreter','latex','FontSize',12);
% Create ylabel
ylabel({'$p(l)$'},'Interpreter','latex','FontSize',12);
% Create title
legend1 = legend('$\l_{min} = 0.5$','$\l_{min} = ...
    1$','$\l_{min} = 2 $',...
     '$\l_{min} = 3 $');
set(legend1,'Interpreter','latex','FontSize',12)

%% Plotting normalized again with keeping l_mean = ...
    constant for diff values
% of mu. The intention is show how the curve looks like ...
    for the range of
% interest.
mu = 3; %1.1, 1.5, 2, 2.5, 3
lmin = 1;
l=[1:0.1:6];
C=(mu-1)*lmin.^(mu-1);
p=C*l.^(-mu);
plot(l,p,':') % '--','.','-',':'
hold on
% Create xlabel
xlabel({'$\l$'},'Interpreter','latex','FontSize',12);
% Create ylabel
ylabel({'$p(l)$'},'Interpreter','latex','FontSize',12);
% Create title
legend1 = legend('$\mu = 1.1$','$\mu  = 1.5$','$\mu  = 2 ...
    $',...
     '$\mu  = 2.5 $','$\mu  = 3 $');
set(legend1,'Interpreter','latex','FontSize',12)

%% log scale
mu = 3; %1.1, 1.5, 2, 2.5, 3
lmin = 1;
l=[0.1:0.1:10]; % 0.5, 1, 2, 3
C=(mu-1)*lmin.^(mu-1);
```

```
p=C*l.^(-mu);
loglog(l,p,'-.') % '--','.','-',':','-.'
hold on
% Create xlabel
xlabel({'$\l$'},'Interpreter','latex','FontSize',11);
% Create ylabel
ylabel({'$p(l)$'},'Interpreter','latex','FontSize',11);
% Create title
legend1 = legend('$\mu = 1.1$','$\mu  = 1.5$','$\mu  = 2 ...
   $',...
    '$\mu  = 2.5 $','$\mu  = 3 $');
set(legend1,'Interpreter','latex')
```

```
%% Plot of Y as lmean and x as mu : rv = 1
clear all;
lambda =100; %10000,1000,100
rv=1;
mu_i=[1.01:0.12:2.99];
mu_length=length(mu)
for i=1:length(mu_i)
mu=mu_i(i);
k1=(mu-1)/(2-mu)
k2=((lambda^(2-mu)) -(rv^(2-mu)))/(rv^(1-mu))
k3=(lambda^(2-mu))/rv^(1-mu)
lmean = (k1*k2)+k3;
lmean_s(i)=lmean;
end
semilogy(mu_i,lmean_s,'-.')
hold on
% Create xlabel
xlabel({'$\mu$'},'Interpreter','latex','FontSize',12);
% Create ylabel
ylabel({'$l_{mean}$'},'Interpreter','latex','FontSize',12);
% Create title
% title({' $lmean versus \mu for r_v=1$'},...
%     'Interpreter','latex',...
%     'FontSize',11);
% Create legend
legend1 = legend('$\lambda = 10^4$','$\lambda = ...
   10^3$','$\lambda = 10^2$');
set(legend1,'Interpreter','latex','FontSize',12)

%% Plot of lmin and rv with a fixed rho.
clear all
lambda = 10; %10000,1000,100,10
rv_i = 0.1:0.1:5;
mu=2.05
```

```
for i = 1:length(rv_i)
rv = rv_i(i);
rho = (2*lambda*rv)^-1;
k1=(mu-1)/(2-mu);
k2=((lambda^(2-mu)) -(rv^(2-mu)))/(rv^(1-mu));
k3=(lambda^(2-mu))/rv^(1-mu);
lmean = (k1*k2)+k3;
lmin = (lmean)*((mu-2)/(mu-1));
lmin_i(i) = lmin;
end
hold on
plot(rv_i,lmin_i,'.')

% Create xlabel
xlabel({'$r_v$'},'Interpreter','latex','FontSize',12);
% Create ylabel
ylabel({'$l_{min}$'},'Interpreter','latex','FontSize',12);
% Create legend
legend1 = legend('$\lambda = 10^4$','$\lambda = ...
    10^3$','$\lambda = 10^2$',...
    '$\lambda = 10$');
set(legend1,'Interpreter','latex','FontSize',12)
legend boxoff
```

```
%% Plotting mu versus efficiency equation(3).
%rv=1/(2*lambda*rho);
set(gcf,'DefaultAxesColorOrder',[0 0 1],...
'DefaultAxesLineStyleOrder','-|--|-.|:')
for lambda = [10000,1000,100,10]
    rv=1;
    mu_i=[1:0.05:3];
    for i=1:length(mu_i)
        mu=mu_i(i);
        k1=(mu-1)/(2-mu);
        k2=((lambda^(2-mu)) -(rv^(2-mu)))/(rv^(1-mu));
        k3=(lambda^(2-mu))/rv^(1-mu);
        lmean = (k1*k2)+k3;
        N = (lambda/rv)^((mu-1)/2);  %Non-destructive N
        eff= 1/(N*lmean);
        l_eff(i) = lambda*eff;
        hold all
    end
    plot(mu_i,l_eff)
end
hold off
xlabel({'$\mu$'},'Interpreter','latex','FontSize',12);
ylabel({'$\lambda\eta$'},'Interpreter','latex','FontSize',12);
```

```
legend1 = legend('$\lambda = 10^4$','$\lambda = ...
    10^3$','$\lambda = 10^2$',...
    '$\lambda = 10$');
set(legend1,'Interpreter','latex','FontSize',12)
%% Effect of lmin on lmean AT a close value of mu.
clear figure; clear all;
set(gcf,'DefaultAxesColorOrder',[0 0 1],...
'DefaultAxesLineStyleOrder','-|--|-.|:')
    for mu = [2.01 2.05 2.1];
        lmin = [0.1:0.1:5];
        lmean = ((mu-1)/(mu-2))*lmin;
        plot(lmin,lmean)
        hold all
    end
    hold off
% Create xlabel
xlabel({'$\l_{min}$'},'Interpreter','latex','FontSize',12);
% Create ylabel
ylabel({'$\l_{mean}$'},'Interpreter','latex','FontSize',12);
% Create legend
legend1 = legend('$\mu = 2.01$','$\mu = 2.05$','$\mu = 2.1$');
set(legend1,'Interpreter','latex','FontSize',12)
```

```
function [total_distance,mu_estimated] = total_distance ...
    (mu,steps)
u = rand(1,1E4); %uniformaly distributed open interval (0,1)
l_min=1;
%mu=1.5;
f=l_min*(1-u).^-(1/(mu-1));
%figure(1);
%hist(f,[1:0.1:10]);

total_distance=0;

%figure(4)
%axis([-1000 1000 -1000 1000])
xi=0;yi=0;
%steps=10000;
n=steps;
log_sum=0;
for i=1:steps
    l=f(1,randi(10000));
    angle=2*pi*rand(1,1);  %uniformaly distributed angle
    total_distance=total_distance+l;
    % xf=xi+l*cos(angle);
    % yf=yi+l*sin(angle);
    % line([xi xf],[yi yf])
    % hold on
    % plot(xf,yf,'o')
    %    xi=xf;
    %    yi=yf;

    log_r = log(l/l_min);
    log_sum = log_sum+log_r;
    mu_estimated = 1+ n*(log_sum)^-1;

end
    total_distance;
end
```

```
clear all, clear figure
steps=[1:300];
for i = 1:length(steps)
[A,B]=total_distance(2,steps(i));
B_i(i)=B;
x_i(i) = mean(B_i);
end
plot(steps,B_i,'.','LineWidth',1)
hold on
plot(steps,x_i,'r-','LineWidth',1.2)
hold off
title('Maximum Likelihood Estimator (MLE) method', ...
    'Interpreter', 'latex','FontSize',12)
xlabel({'$n$ (The count of sample lengths)'},'Interpreter'...
    ,'latex','FontSize',12);
% Create ylabel
ylabel({'$\mu_{est}$ \,(The estimated ...
    $\mu$)'},'Interpreter',...
    'latex','FontSize',12);
% Create legend
 legend1 = legend('$\mu_{est}$','mean($\mu_{est}$)');
set(legend1,'Interpreter','latex','FontSize',12, ...
    'box','on',...
    'Xcolor',[1 1 1],'Ycolor',[1 1 1]);
% grids
set(gca,'YLim',[1.5 3])
set(gca,'YTick',[1:0.1:4])
grid on
%set(gca,'YTickLabel',['0';' ';'1';' ';'2';' ';'3';' ';'4'])
```

```
clear figure
     set(gcf,'DefaultAxesColorOrder',[0 0 1],...
'DefaultAxesLineStyleOrder','-|--|-.|:')
```

```
for ep = [0.1,0.5,1]
    lmin = [0.5:0.1:10];
    l = lmin+ep;
    mu=2.01;
    CDF= 1-(l./lmin).^(1-mu);

    plot(lmin,CDF)
    hold all
end
xlabel({'$l_{min}+\epsilon$'},'Interpreter','latex',...
    'FontSize',12);
% Create ylabel
ylabel({'Cumulative Density Fcn'},'Interpreter','latex',...
    'FontSize',12);
% Create legend
legend1 = legend('$\epsilon = 0.1 $','$\epsilon = 0.5 $',...
    '$\epsilon = 1 $');

set(legend1,'Interpreter','latex','FontSize',12, ...
   'box','on',...
    'Xcolor',[1 1 1],'Ycolor',[1 1 1]);
set(gca,'XLim',[0 10],'FontSize',10,'XTick',[0:1:10])
grid on
```

```
% lmean versus rv
clear all;
rv_i=[0.01:0.05:5];
rho = 0.1; %1, 0.1
mu=2.01;
for i =1:length(rv_i)
rv=rv_i(i);
k1 = (mu-1)/(mu-2);
k2=(2*rho)^(mu-2);
l_mean= ((-k1*k2+k2)*(rv.^(2*mu-3)))+(k1*rv);
l_mean_i(i)= l_mean;
end
plot(rv_i,l_mean_i,'-') % --, -
hold on
xlabel({'$r_v$'},'Interpreter','latex','FontSize',12);
ylabel({'$l_{mean}$'},'Interpreter','latex','FontSize',12);
legend1 = legend('$\rho = 1, \mu = 2.01$','$\rho = 0.1, ...
   \mu = 2.01$');
set(legend1,'Interpreter','latex','FontSize',12, ...
   'box','on',...
    'Xcolor',[1 1 1],'Ycolor',[1 1 1]);
set(gca,'XTick',[0:0.5:5])
grid on
```