Multi-Armed Bandit (at intern), Electromagnetic Modeling

A PROJECT REPORT

*Submitted in partial fulfillment of the
requirements for the award of the degrees*

of
BACHELOR OF TECHNOLOGY
in

ELECTRICAL ENGINEERING

Submitted by:
KUNAL YADAV

Guided by:
Dr. SAPTARSHI GHOSH

INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2019

CANDIDATE’S DECLARATION

I hereby declare that the project entitled **Multi-Armed Bandit, Electromagnetic Modeling** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Electrical Engineering completed under the supervision of **Dr. Saptarshi Ghosh, Assistant Professor, Department of Electrical Engineering** IIT Indore is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Kunal Yadav

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the students is correct to the best of my knowledge.

Dr Saptarshi Ghosh

Assistant Professor

Department of Electrical

Engineering IIT Indore

Preface

This report on **Multi-Armed Bandit, Electromagnetic Modeling** is prepared under the guidance of Dr. Saptarshi Ghosh.

Though this report my work at intern at Games 24x7 a customized Multi-Armed Bandit with various Algorithms and Reward functions is explained with the help of various tables and graphs. Also, the Project that I did using the skills acquired at intern that is Electromagnetic Modeling is explained. In Electromagnetic Modeling a Multilayer Electromagnetic with desired properties was designed.

Kunal Yadav
B.Tech. IV Year
Discipline of Electrical Engineering
IIT Indore

Acknowledgements

We wish to thank Dr Saptarshi Ghosh for his kind support and valuable guidance.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Kunal Yadav
B.Tech. IV Year
Discipline of Electrical Engineering
IIT Indore

Abstract

While Testing for new feature you need to explore the performance of various different features to find out the feature that gives the maximum output and then you exploit the best feature to attain maximum reward. Due to exploration of underperforming features a regret is generated and to minimize this regret Multi Armed Bandit balances the trade-off between exploration and exploitation, and it helps us to find the best path such as regret in minimized during the process of finding the best features.

Electromagnetic broadband, thin, and lightweight absorbers are getting increasing interest in both civil and military applications. Accordingly, the need for high-performance absorbing structures has prompted the need for conceiving and manufacturing tailored materials with very low specular reflection and transmission. This project is to build a Multi-layer electromagnetic wave absorber with minimum thickness and minimum reflectivity, Over a broad band of frequency. To achieve desired results a customized reward function is used. And then we maximize the customized reward function to find the best possible combination of layers to build Multi-layer electromagnetic wave absorber.

Table of Contents

List of Figures

List of Tables

Introduction

Exploration vs Exploitation:

Exploration:

Exploration is required to find out the true potential of reward. If you don’t explore you won't find out whether the path you are going through give the maximum reward. But if you do much exploring you would be increasing the regret by diverting more traffic to underperforming paths.

Exploitation:

Exploitation is diverting the entire traffic to the path that was previously determined to be best performing path with the help of previously performed experiments or survey. To attain maximum instantaneous reward exploitation is required.

Tradeoff Exploitation vs Exploration:

Exploration is required to find out what is the true potential of reward. whereas one must exploit to best performing path to attain the maximum reward. One must balance between exploration and exploitation to obtain maximum possible reward.

We need some Algorithm to decide when to explore and when to exploit also we need to decide what part of traffic to explore and what part of traffic to exploit.

On-line decision making involves a fundamental choice; exploration, where we gather more information that might lead us to better decisions in the future or exploitation, where we make the best decision given current information.

Exploration and Exploitation are both required to find the maximum reward but we must balance the tradeoff between exploration and exploitation to maximize the reward in the process of finding the best path.

Multi-Armed bandit:

In the well-studied multi-armed bandit problem, a gambler must choose which of K slot machines to play. At each time step, he pulls the arm of one of the machines and receives a reward or payoff (possibly zero or negative). The gambler’s purpose is to maximize his total reward over a sequence of trials. Since each arm is assumed to have a different distribution of rewards, the goal is to find the arm with the best expected return as early as possible, and then to keep gambling using that arm. The problem is a classic example of the trade-off between exploration and exploitation. On the one hand, if the gambler plays exclusively on the machine that he thinks is best (“exploitation”), he may fail to discover that one of the other arms actually has a higher average return. On the other hand, if he spends too much time trying out all the machines and gathering statistics (“exploration”), he may fail to play the best arm often enough to get a high total return.

As a part of my project at internship a customized Multi-Armed Bandit framework was developed that can be used for both testing of previously performed experiments and also divert the live traffic and decide when to explore and exploit to maximize the factors such as revenue and user converted. The framework included combination of various Multi-Armed Bandit Algorithms and various reward functions. The framework is better than the classical AB Testing because the framework minimizes the regret caused due to sending more traffic to underperforming paths.

Reward Functions used:

The following Reward functions were used during the project:

- Binomial Reward
- Average Revenue Per User
- Weighted Ranks

Each of the above Reward functions with their significance would be explained in subsequent chapters.

Algorithms used:

The following Algorithms were used during the project:

- Epsilon-Greedy
- Epsilon-Greedy with Annealing
- Softmax

Each of the above Algorithms with their performance would be explained in subsequent chapters.

Reward Functions

Binomial Reward:

Binomial reward is just to check whether a user give any revenue or not. It is used to check a feature that is responsible to increase a customer base of a platform. Basically, the binomial reward is equal to the number of conversions. For example, assume a company wants to update the design of the signup page and the designers come up with 3 designs assuming each design as a path, the reward per path is equal to the number of users signing up after visiting that path divided by the number of users visiting that path.

The binomial reward of a particular path does not depend on the amount of revenue obtained by the users.

Steps to calculate Binomial Reward:

1. Count the total number of users passed through the given path.
2. Count the number of users that generate revenue
3. Calculate the fraction of users that generate revenue.
4. The faction is the Binomial reward of a given path.

Limitations of Binomial Reward:

The binomial reward cannot determine how much a feature encourages users to spend, When we only consider binomial reward the reward of path in which users of less revenue would be more than the path with less users with more revenue, So there might be a possibility that the overall revenue of the winning path might not be the best.

Average Revenue Per User:

Average Revenue per User is the average off the revenue generated per user. It eliminates the possibility that the overall revenue of the winning path might not be the best this was the major problem faced in Binomial Reward Function.

The Significance of Average Revenue per User reward is that the path generating maximum average revenue would have maximum reward.

Steps to calculate Average Revenue Per User Reward:

1. Count the total number of users passed through the given path.
2. Calculate the sum of revenue generated by all users.
3. Calculate the fraction of total revenue and total number of users that generate revenue.
4. The faction is the Average Revenue Per User reward of a given path.

Limitations of Average Revenue Per User Reward:

The Average Revenue Per User reward fluctuates a lot because of a overperforming or a underperforming individual so it might give maximum instantaneous revenue but in a long run it might not be a best performing path. So Average Revenue Per User reward function is not the best reward function for long run.

Weighted Rank Reward:

In weighted Rank Reward function ranks of the revenue is ranks than the ranks are multiplied with weights based on the slots in which the revenue lies. Then average is taken to calculate the reward.

The significance of Weighted Rank Reward function is that it maximum the revenue while taking care of overperforming and underperforming individuals so in a long run weighted rank reward function is the best reward function.

Steps to calculate Average Revenue Per User Reward:

1. Count the total number of users passed through the given path.
2. Calculate the rank of revenue generated by users.
3. Calculate the product of ranks and weights based on the slots in which revenue lies.
4. Calculate the sum of products of ranks and weights.
5. Calculate the fraction of sum of products of ranks and weights and total number of users.
6. The faction is the Weighted Rank Reward of a given path.

Limitations of Average Revenue Per User Reward:

The only Limitation of Weighted rank reward function is the computation time, it requires more computation time as compared with other reward functions.

Comparison of Reward Functions:

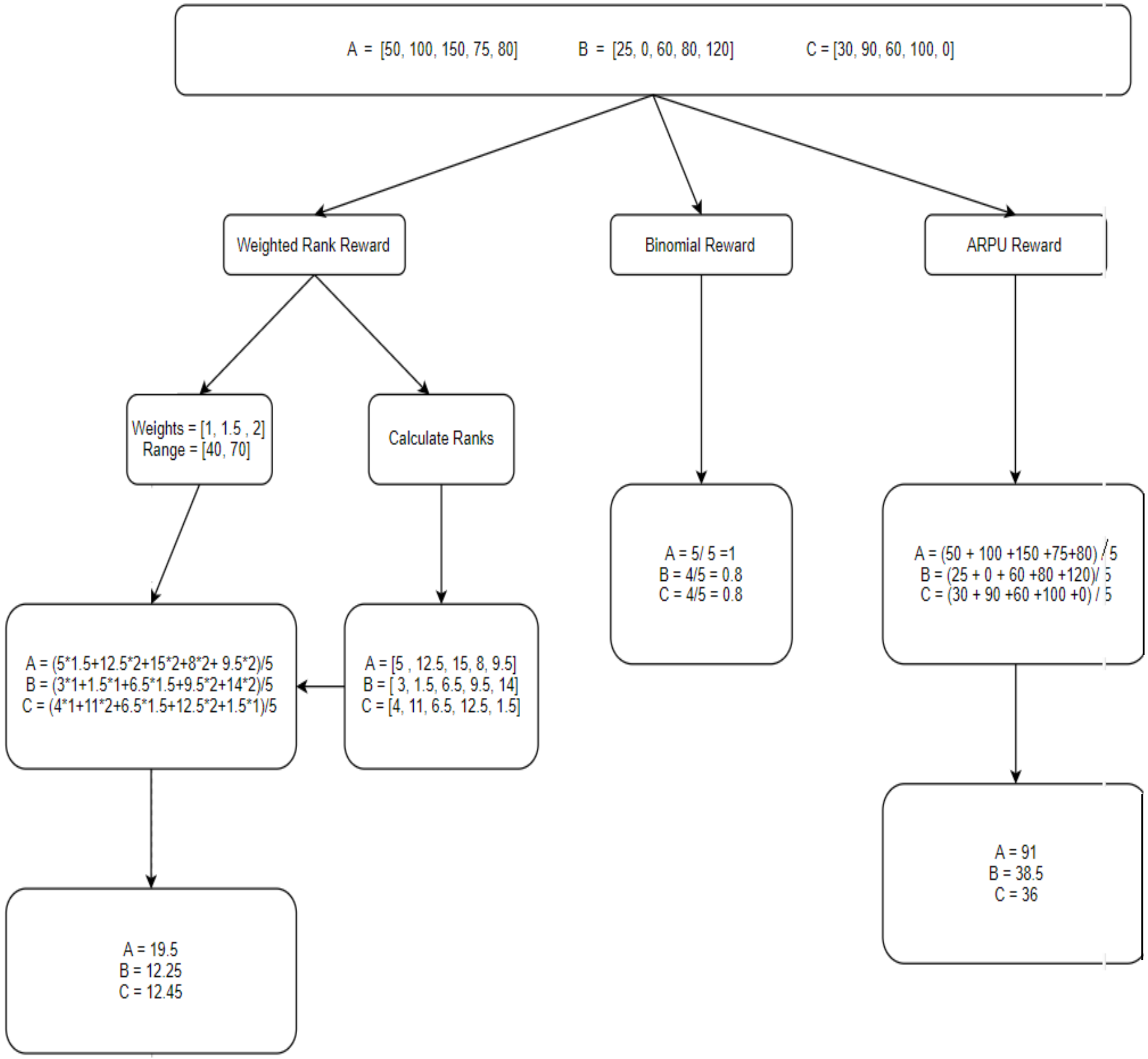
A = [50, 100, 150, 75, 80]
B = [25, 0, 60, 80, 120]
C = [30, 90, 60, 100, 0]
Weighted Rank Reward
Binomial Reward
ARPU Reward
Weights = [1, 1.5 , 2]
Range = [40, 70]
Calculate Ranks
A = 5/ 5 =1
B = 4/5 = 0.8
C = 4/5 = 0.8
A = (50 + 100 +150 +75+80) /5
B = (25 + 0 + 60 +80 +120)/ 5
C = (30 + 90 +60 +100 +0) / 5
A = (5*1.5+12.5*2+15*2+8*2+ 9.5*2)/5
B = (3*1+1.5*1+6.5*1.5+9.5*2+14*2)/5
C = (4*1+11*2+6.5*1.5+12.5*2+1.5*1)/5
A = [5 , 12.5, 15, 8, 9.5]
B = [3, 1.5, 6.5, 9.5, 14]
C = [4, 11, 6.5, 12.5, 1.5]
A = 91
B = 38.5
C = 36
A = 19.5
B = 12.25
C = 12.45

Fig1 Comparison of Reward functions

Algorithms Used

Epsilon Greedy:

The ε -greedy algorithm is widely used because it is very simple, and has obvious generalizations for sequential decision problems. At each round t = 1, 2, ... the algorithm selects the arm with the highest empirical mean with probability 1 − ε, and selects a random arm with probability ε. In other words, given initial empirical means µ1(0), ..., µK (0).

$$p_i(t+1) = \begin{cases} 1 - \epsilon + \epsilon/k & \text{if } i = \arg\max_{j=1,\ldots,K} \hat{\mu}_j(t) \\ \epsilon/k & \text{otherwise.} \end{cases}$$

Here ε decide the rate of exploration the rate of exploration is directly proportional to the value of ε.

Epsilon Greedy with Annealing:

In ε -greedy with annealing ε is varied over time. As the ε decreases the rate off exploration decreases and the algorithm exploits more as time step progresses. In case ε -greedy without annealing only a linear bound on the expected regret can be achieved as ε is held constant.

Since exploration is directly proportional to ε the algorithm starts divert more traffic to better performing path over time.

Softmax:

Softmax methods are based on Luce’s axiom of choice (1959) and pick each arm with a probability that is proportional to its average reward. Arms with greater empirical means are therefore picked with higher probability. In our experiments, we study Boltzmann exploration, a Softmax method that selects an arm using a Boltzmann distribution. Given initial empirical means µ1(0), ..., µK(0).

$$p_i(t+1) = \frac{e^{\hat{\mu}_i(t)/\tau}}{\sum_{j=1}^{k} e^{\hat{\mu}_j(t)/\tau}}, i = 1 \ldots n$$

where τ is a temperature parameter, controlling the randomness of the choice. When τ = 0, Boltzmann Exploration acts like pure greedy. As τ tends to infinity, the algorithms picks arms uniformly at random.

Comparison of Epsilon greedy and Softmax:

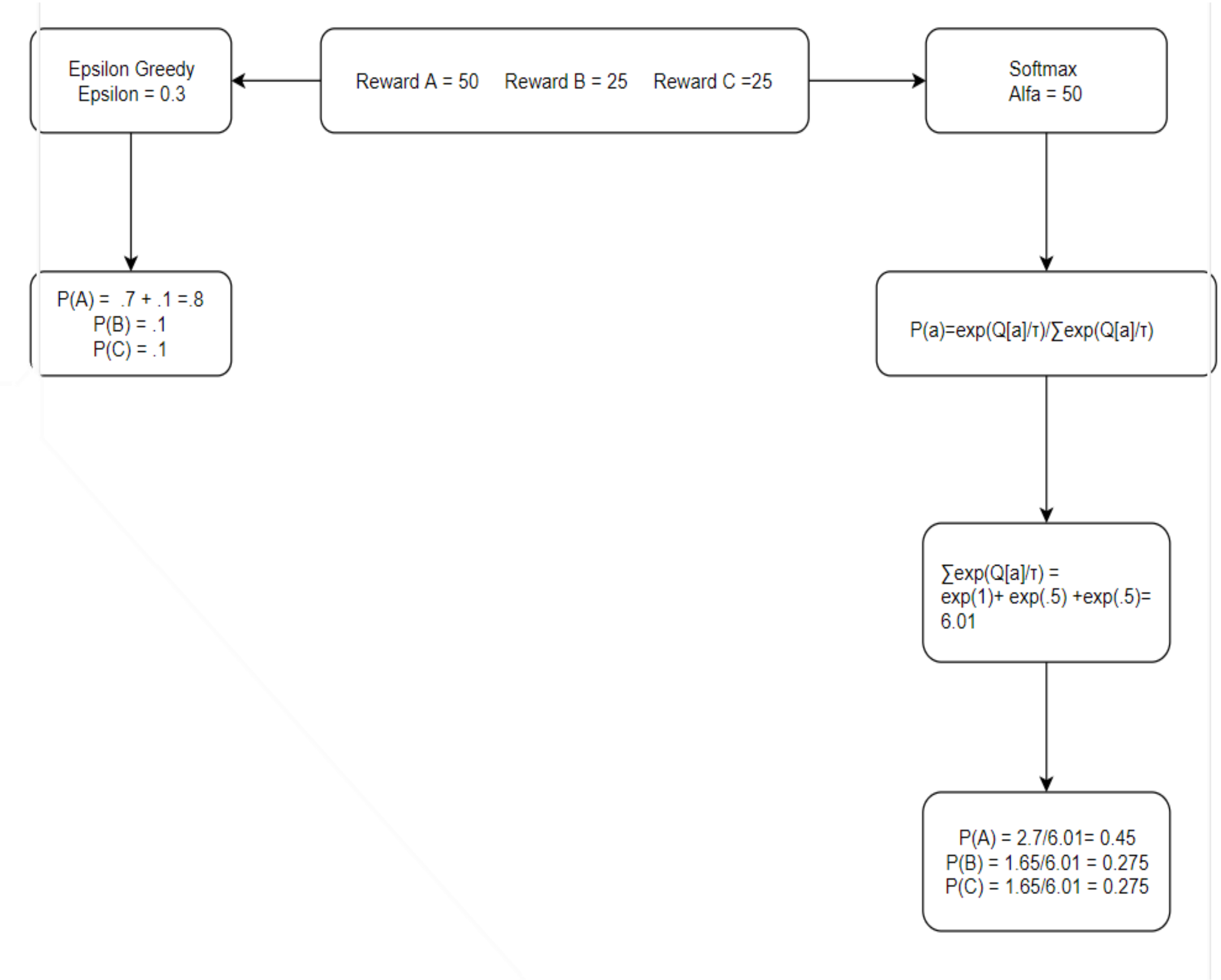

Fig2 Comparison of Epsilon Greedy and Softmax

Experiment and Results

Experiment:

- In this project different combination of Multi-Armed Bandit Algorithms were used along with different reward functions.
- In the initial time steps that is first 5 days for the experiment pure exploration is done that is the traffic is diverted equally through all the paths.
- Then after 5 days the revenue generated by a user is taken in account and then it is provided to the Algorithm.
- The algorithm first calculates the reward based on revenue data.
- Then the algorithm decides how to split the traffic based on the Multi armed bandit algorithm used.
- This project was carried on live data obtained during my internship.
- But for the sake of companies NDA the experiment is carried on a lognormally distributed data.
- And thousand simulations of the same in done and then the results are shown as a average of thousand simulation.
- Lognormal distribution is used because it was the most similar distribution of the real data.

Results:

Epsilon Greedy Binomial reward:

The following results were obtained when the experiment was done on 2 paths with revenue distributed lognormally with conversion of 80 percent and 70 percent respectively.

day	traffic path 1	Overall conv	traffic path 2	Overall conv
1	505	0.81	495	0.69
2	499	0.79	501	0.71
3	495	0.78	505	0.72
4	508	0.82	492	0.71
5	502	0.77	498	0.68
6	752	0.84	248	0.7
7	748	0.8	252	0.72
8	755	0.81	245	0.71
9	745	0.8	255	0.7
10	752	0.79	248	0.68
11	744	0.81	256	0.71
12	749	0.8	251	0.69
13	750	0.81	250	0.7
14	752	0.79	248	0.71
15	748	0.8	252	0.7

Table 1 Epsilon Greedy Binomial reward

The traffic split of the same is shown below.

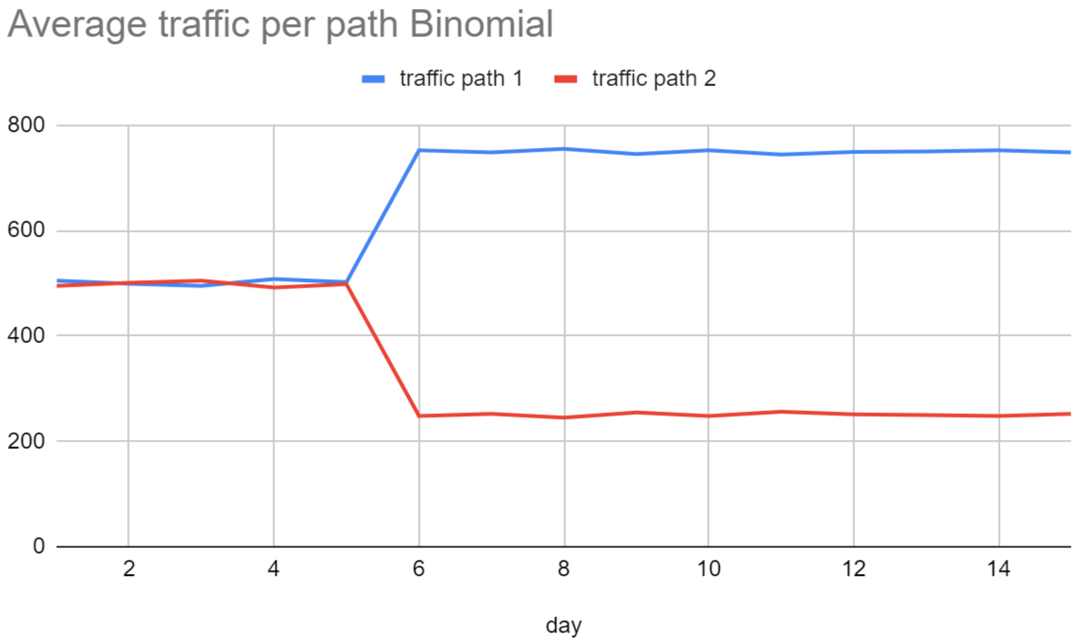

Fig3 Epsilon greedy binomial reward

It could be observed from the above graph that average traffic for both the paths are same for first 5 days the initial phase when exploration is done then the traffic is constant with path 1 being exploited more, the same was expected from the results since the conversion rate of path 1 was more than that of path 2.

Epsilon Greedy Weighted Rank Reward:

The following results were obtained when the experiment was done on 2 paths with revenue distributed lognormally with mean of 4.2 and 4.4 respectively and variance of 2.14 and a conversion rate of 80 percent for both.

day	traffic path 1	Reward p1	traffic path 2	Reward p2
1	495	100.22	505	85.20
2	501	108.33	499	88.31
3	505	121.23	495	89.41
4	492	130.94	508	91.85
5	498	141.44	502	93.96
6	752	151.95	248	96.06
7	748	162.45	252	98.17
8	758	172.96	242	100.27
9	744	183.46	256	102.38
10	739	193.97	261	104.48
11	757	204.47	243	106.59
12	760	214.98	240	108.69
13	748	225.48	252	110.80
14	743	235.99	257	112.90
15	755	246.49	245	115.01

Table 2 Epsilon Greedy Weighted rank reward

The traffic split of the same is shown below.

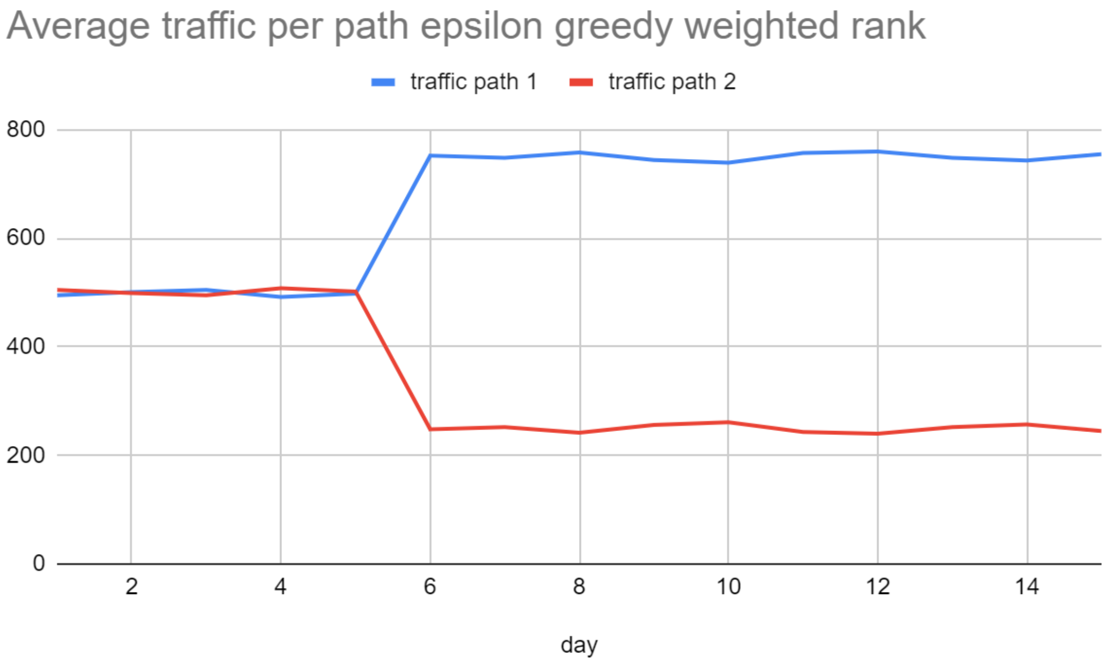

Fig4 Epsilon greedy Weighted rank reward

It could be observed from the above graph that average traffic for both the paths are same for first 5 days the initial phase when exploration is done then the traffic is constant with path 1 being exploited more, the same was expected from the results since the Weighted rank reward of path 1 was more than that of path 2.

Epsilon Greedy Average Revenue Per User Reward:

The following results were obtained when the experiment was done on 2 paths with revenue distributed lognormally with mean of 4.2 and 4.4 respectively and variance of 2.14 and a conversion rate of 80 percent for both.

day	traffic path 1	Reward p1	traffic path 2	Reward p2
1	489	55.80	511	44.50
2	512	58.62	488	45.10
3	503	57.83	497	44.80
4	495	56.25	505	44.90
5	506	57.10	494	45.10
6	751	58.19	249	45.15
7	748	58.21	252	45.30
8	753	58.24	247	45.43
9	744	58.36	256	45.55
10	745	58.38	255	45.68
11	753	58.43	247	45.80
12	742	58.49	258	45.93
13	740	58.54	260	46.05
14	749	58.59	251	46.18
15	757	58.65	243	46.30

Table 3 Epsilon Greedy ARPU reward

The traffic split of the same is shown below.

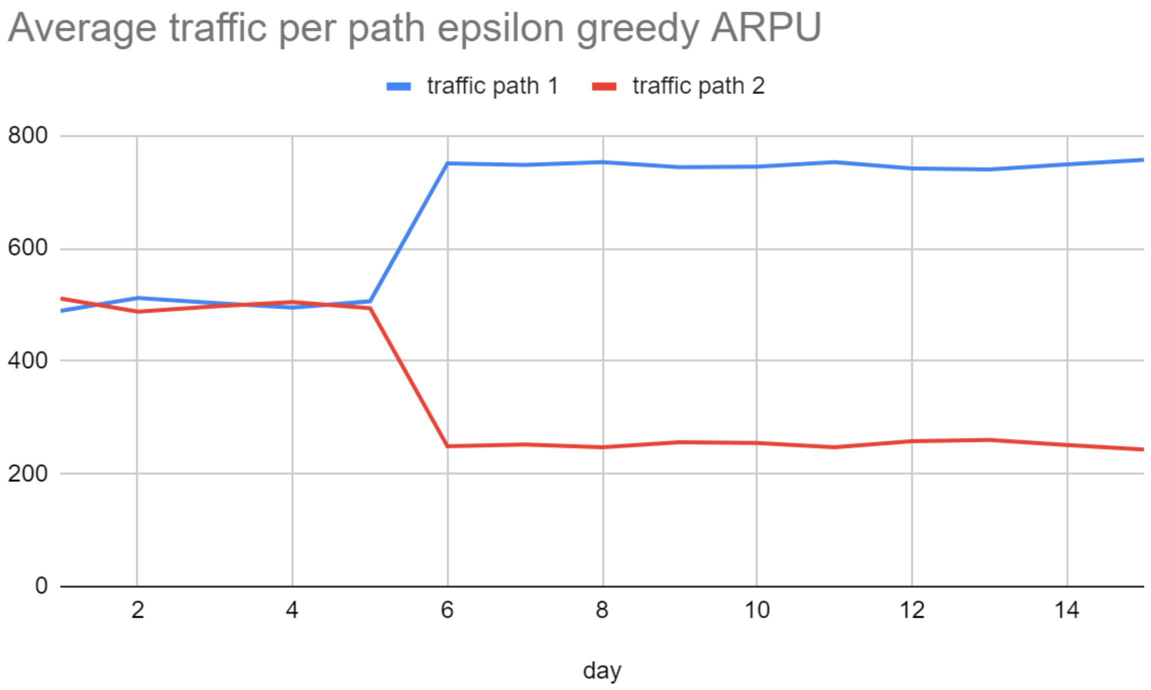

Fig5 Epsilon greedy ARPU reward

It could be observed from the above graph that average traffic for both the paths are same for first 5 days the initial phase when exploration is done then the traffic is constant with path 1 being exploited more, the same was expected from the results since the Average Revenue of path 1 was more than that of path 2.

It could also be observed that traffic split for all three reward functions is identical this is because in epsilon greedy traffic split doesn’t depend on value of reward.

Epsilon Greedy with annealing Weighted Rank Reward:

The following results were obtained when the experiment was done on 2 paths with revenue distributed lognormally with mean of 4.2 and 4.4 respectively and variance of 2.14 and a conversion rate of 80 percent for both.

day	traffic path 1	Reward p1	traffic path 2	Reward p2
1	489	108.33	511	75.20
2	512	121.23	488	78.31
3	503	130.94	497	79.41
4	495	141.44	505	81.85
5	506	152.75	494	83.96
6	751	163.65	249	86.06
7	765	174.55	235	88.17
8	775	185.46	225	90.27
9	782	196.36	218	92.38
10	794	207.27	206	94.48
11	804	218.17	196	96.59
12	815	229.07	185	98.69
13	825	239.98	175	100.80
14	835	250.88	165	102.90
15	846	261.79	154	105.01

Table 4 Epsilon Greedy with annealing weighted rank reward

The traffic split of the same is shown below.

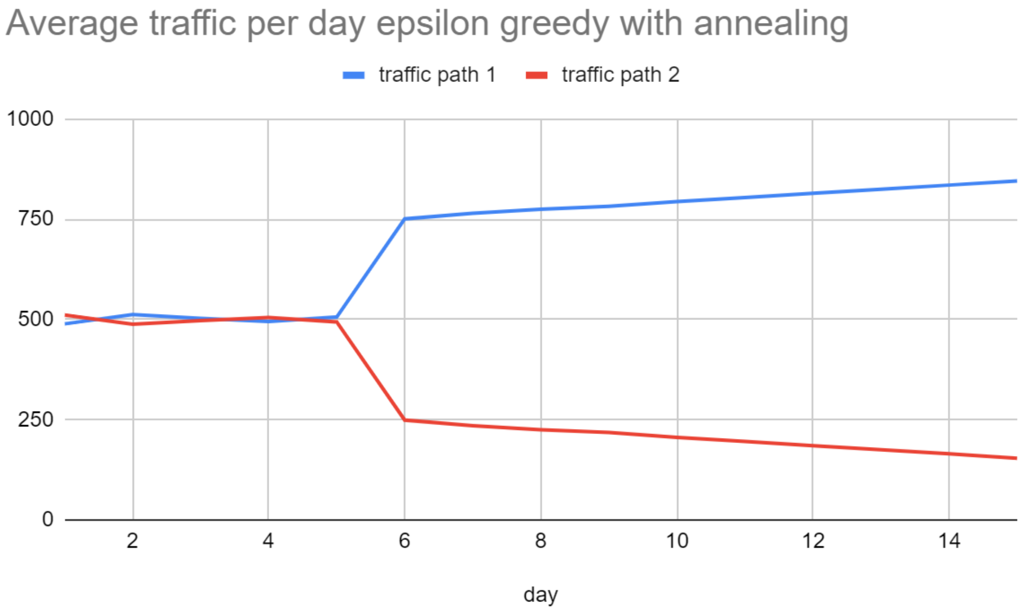

Fig6 Epsilon greedy with annealing weighted rank reward

It could be observed from the above graph that average traffic for both the paths are same for first 5 days the initial phase when exploration is done then the traffic is constant with path 1 being exploited more, the same was expected from the results since the Weighted rank reward of path 1 was more than that of path 2.

It could also be observed that the rate of exploration decreases over time and the traffic for path 1 increases with each time step.

Epsilon Greedy Weighted Rank Reward:

The following results were obtained when the experiment was done on 2 paths with revenue distributed lognormally with mean of 4.2 and 4.4 respectively and variance of 2.14 and a conversion rate of 80 percent for both.

day	traffic path 1	Reward p1	traffic path 2	Reward p2
1	489	121.23	511	81.85
2	512	130.94	488	83.96
3	503	144.44	497	89.06
4	495	155.41	505	92.16
5	506	167.02	494	95.77
6	751	178.63	249	99.37
7	782	190.23	218	102.98
8	800	201.84	200	106.59
9	836	213.44	164	110.19
10	861	225.05	139	113.80
11	888	236.66	112	117.40
12	915	248.26	85	121.01
13	942	259.87	58	124.61
14	970	271.47	30	128.22
15	987	283.08	13	131.82

Table 5 Softmax weighted rank reward

The traffic split of the same is shown below.

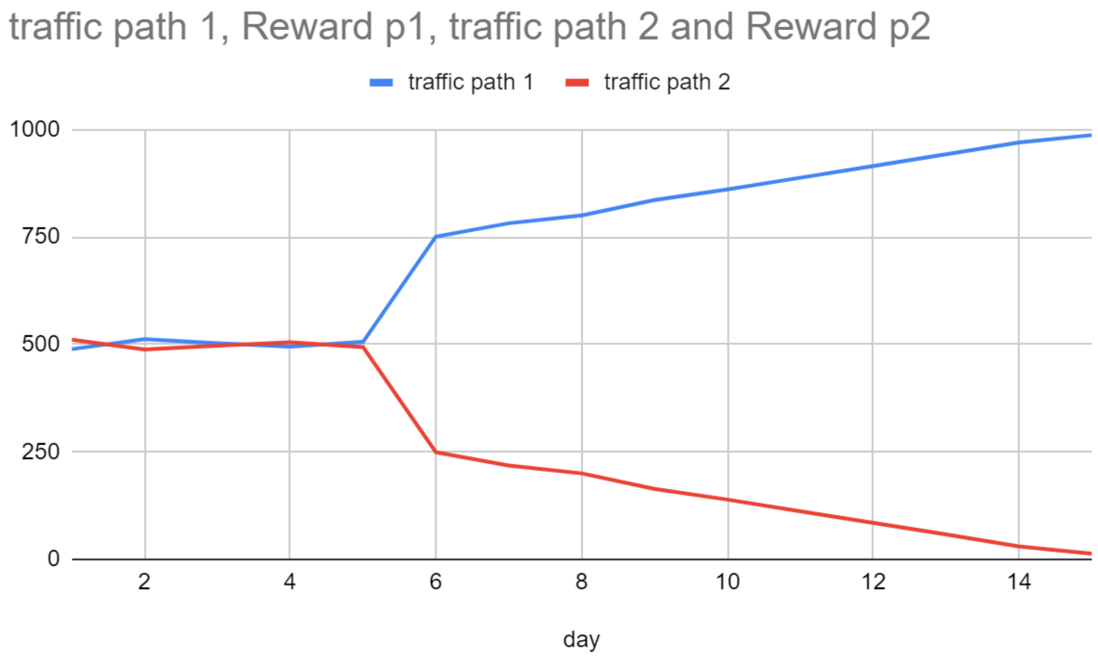

Fig7 Softmax weighted rank reward

It could be observed from the above graph that average traffic for both the paths are same for first 5 days the initial phase when exploration is done then the traffic is constant with path 1 being exploited more, the same was expected from the results since the Weighted rank reward of path 1 was more than that of path 2.

It could also be observed that the rate of exploration decreases over time and the traffic for path 1 increases as the ratio of reward increases with time.

Also, it could be observed that the minimum regret is obtained in softmax as the maximum traffic of path 1 in the end of experiment is maximum in case of softmax.

Multilayer Electromagnetic Wave Absorber

Introduction:

This project is to build a Multi-layer electromagnetic wave absorber with minimum thickness and minimum reflectivity, Over a broad band of frequency. To achieve desired results a customized reward function is used. And then we maximize the customized reward function using Optimization toolbox in MATLAB to find the best possible combination of layers to build Multi-layer electromagnetic wave absorber.

Electromagnetic Absorbers:

Electromagnetic absorbers are specifically chosen or designed materials that can inhibit the reflection or transmission of electromagnetic radiation. For example, this can be accomplished with materials such as dielectrics combined with metal plates spaced at prescribed intervals or wavelengths. The particular absorption frequencies, thickness, component arrangement and configuration of the materials also determine capabilities and uses.

MULTI LAYER ABSORBER:

Because of little electromagnetic parameters for adjustment Single layer absorbers have the disadvantages of narrow frequency band and thick structure. Fig1 represents the multi-layered coating backed by perfectly conducting ground plane or metal sheet. The wave absorber coating is composed of layers of lossy material. It may be lossy dielectric and/or lossy magnetic material having different characteristic as a function of frequency

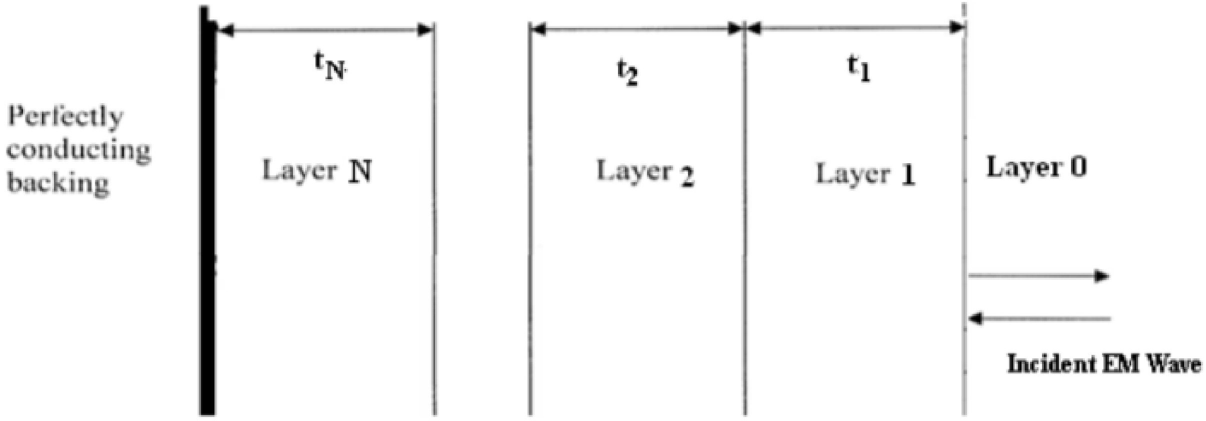

Fig8 Multi-layer electromagnetic wave absorber

Reward function:

Reward Function is chosen such that the main focus is to minimize the thickness with (reflectivity)^2 less than 0.1. The change in reward function due to reflectivity is significantly low when compared due thickness.

Reward Function Thickness:

The function used for thickness factor of reward function is 1-2log(x)/x. The reason behind using this function is because a slight change in thickness will result in exponential change in reward so reward would be majorly affected by thickness only.

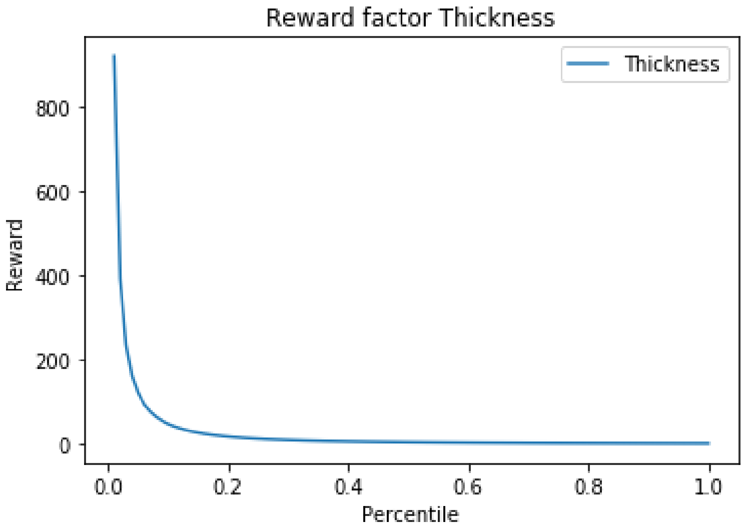

Fig 9 Reward factor thickness

Reward Function Reflectivity:

The function used for reflectivity factor of reward function is 1+sq_root(1-x^2). The reason behind using this function is because a slight change in thickness will result in gradual change in reward so reward would not be much affected by change in reflectivity. But the reward would be zero if reflectivity square is more than 0.1.

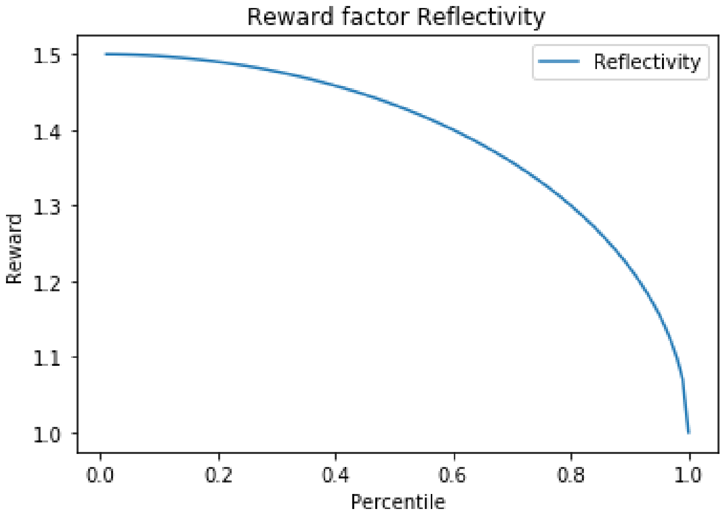

Fig 10 Reward factor reflectivity

Comparison of Reward Functions:

It is clear by comparing both graphs that overall reward would be majorly affected by thickness only.

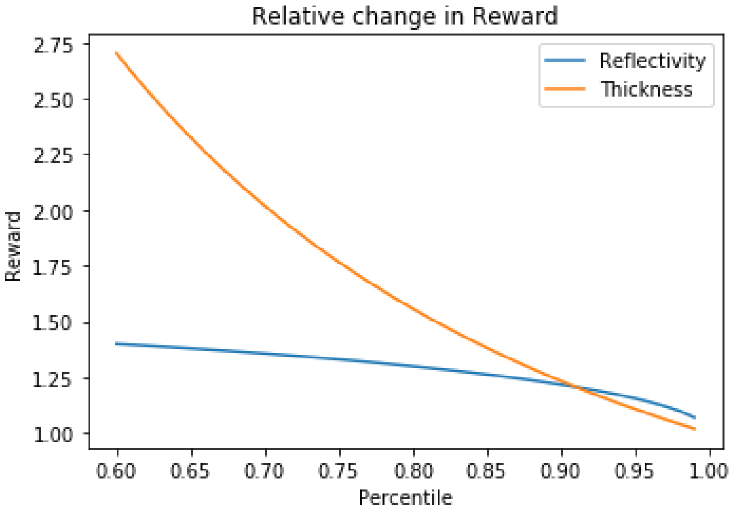

Fig 11 Comparison of reward function

Steps to Calculate reward at a given Combination of thickness:

- Calculation of Reflectivity using a MATLAB code for a range of Frequency 5GHz to 15GHz.
- If the maximum Reflectivity in the given range is greater than Sqrrt(.1) then reward = 0.
- Else Avg Reflectivity is calculated and Reflectivity factor of reward is calculated using Avg Reflectivity.
- Thickness factor is calculated using sum of thickness of all layers.
- Reward is the product of Reflectivity factor Thickness factor.

Properties to materials used:

Following materials were used for the experiment:

	ε r	**ε i**	**μr**	**μi**
Material 1	7.08	0.36	1.92	1.15
Material 2	5.63	2.41	0.12	2.5
Material 3	6.485	3.816	1.893	0.04
Material 4	4.844	1.053	1.529	1.01
Material 5	6.48	0.333	2.05	1.81
Material 6	3.325	13.65	1	0

Table 6 properties of material used

Results:

The following results were obtained for a two-layer absorber:

Layer 1	Thickness 1 (mm)	Layer 2	Thickness 2 (mm)	Reward
1	0.3	2	1.93	8.92
1	0.35	3	1.76	9.1
1	0.421	4	1.89	8.71
1	0.52	5	1.2	9.32
1	0.81	6	1.99	8.55
2	1.301	3	0.902	7.8
2	0.891	4	1.201	9.24
2	1.12	5	0.732	9.02
2	1.05	6	0.991	8.3
3	**0.605**	**4**	**0.721**	**10.41**
3	0.651	5	0.932	9.72
3	0.781	6	1.3	8.98
4	1.88	5	0.52	10.16
4	1.5	6	0.71	9.62
5	1.72	6	0.4	8.98

Table 7 Result for a two-layer absorber

For a two-layer absorber Maximum reward was obtained for a combination of Material 3 and Material 4 with a thickness of 0.605mm and 0.721mm the reflectivity vs frequency curve for the same is shown in figure bellow.

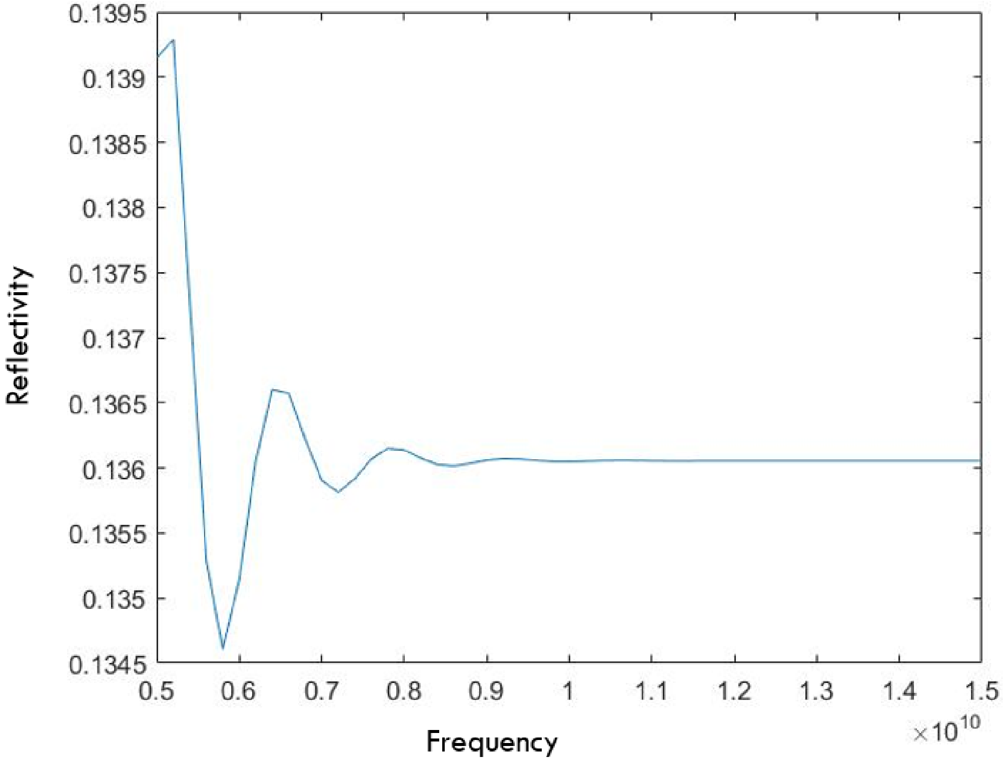

Fig 12 reflectivity vs frequency 2-layer absorber

For a 3-layer absorber maximum reward was obtained for a combination of Material 3, Material 5 and Material 6 with a thickness of 0.426mm ,0.384mm and 0.532mm respectively, the reflectivity vs requency curve for the same is shown in Figure below

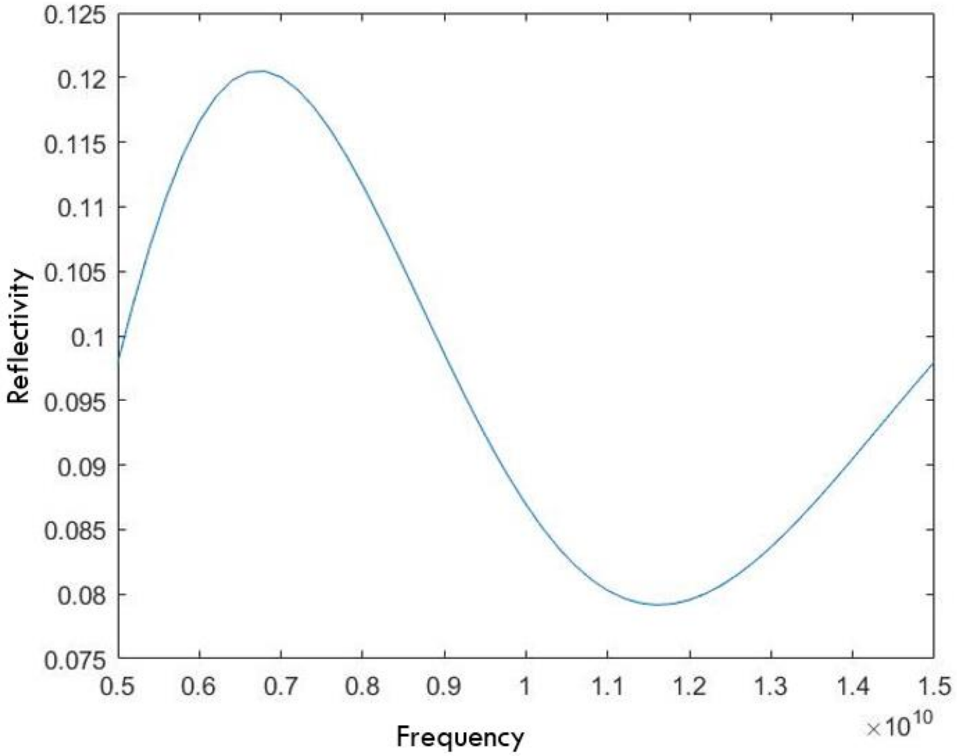

Fig 13 reflectivity vs frequency 3-layer absorber

Conclusion

1. It was observed that softmax outperformed the other algorithms the minimum regret and maximum reward was obtained in case of softmax.

2. The probability distribution of various path in case of epsilon does not depend on the magnitude of reward.

3. Weighted Rank reward function is the best Reward function for long run.

4. For a two-layer absorber Maximum reward was obtained for a combination of Material 3 and Material 4 with a thickness of 0.605mm and 0.721mm.

5. For a 3-layer absorber maximum reward was obtained for a combination of Material 3, Material 5 and Material 6 with a thickness of 0.426mm ,0.384mm and 0.532mm respectively.

References

1. https://towardsdatascience.com/reinforcement-learning-demystified-exploration-vs-exploitation-in-multi-armed-bandit-setting-be950d2ee9f6

2. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=492488

3. H. Robbins. Some aspects of the sequential design of experiments. Bulletin American Mathematical Society, 55:527-535,1952.

4. Volodimir G. Vovk. Aggregating strategies. In Proceedings of the Third Annual Workshop on Computational Learning Theory, pages 371-383,1990.

5. https://arxiv.org/pdf/1402.6028.pdf

6. https://link.springer.com/content/pdf/10.1007/11564096_42.pdf

7. https://ieeexplore.ieee.org/abstract/document/5963741