Sparse Bandit Algorithms for Non-Contiguous Channel Selection for AIoT Networks

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

MICROWAVE AND OPTICAL COMMUNICATION ENGINEERING

Submitted By

Shruti Dwivedi

Under the supervision of

Dr. Rohit Kumar

(Assistant Professor, DTU)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2022

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) Bawana Road,
Delhi-110042

CANDIDATE'S DECLARATION

I, **Shruti Dwivedi (2K20/MOC/07)** student of **M.Tech. (Microwave and Optical Communication Engineering),** hereby declare that the project Dissertation titled "Sparse Bandit Algorithms for Non-Contiguous Channel Selection for AIoT Networks**"** which is submitted by me to the Department of Electronics and Communication Engineering, **Delhi Technological University**, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **SHRUTI DWIVEDI**

Date: 31/05/2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **"Sparse Bandit Algorithms for Non-Contiguous Channel Selection for AIoT Networks"** which is submitted by **Shruti Dwivedi, (2K20/MOC/07)** in Department of Electronics and Communication Engineering, **Delhi Technological University**, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **Dr. ROHIT KUMAR**

Date: 31/05/2022 (SUPERVISOR)

ACKNOWLEDGEMENT

The success and final outcome of this project require a lot of guidance and assistance from many people and I am extremely fortunate to get this all along the completion of my project work. Whatever I have done is only due to such guidance and assistance and I would not forget to thank them.

I owe my profound gratitude to my project guide **Dr. Rohit Kumar** for giving an opportunity to do this work. **Dr. Rohit Kumar** who took keen interest on my project work and guided me throughout, till the completion of my project work by providing all the necessary information, constant encouragement, sincere criticism and sympathetic attitude.

SHRUTI DWIVEDI

ABSTRACT

Wireless communication networks require smart technique to discover resources in restricted shared non-contiguous spectrum in order to make large-scale AIoT a reality. Wideband spectrum analyzer, based on sub-Nyquist sampling and used in Artificial Intelligence of Things (AIoT) gateway, solves this problem. Because the nature of the channels available to us is noncontiguous, so understanding of their occupancy is required. The multi play multi armed bandit (MP-MAB) algorithm is used to model problem of selection of subset.

In this project, we show the ability of learning of such a task using several machine learning algorithms, with a subset having K channels within it, that leads to no reconstruction failure.

Here we observe the comparison among five algorithms which are as follows:

- K subset learning with UCB
- K subset learning with SUCB having fixed sparsity
- K subset learning with SUCB having variable sparsity
- K subset learning with Thompson
- K subset learning with Sparse Thompson

K subset learning with SUCB having variable sparsity and K subset learning with Sparse Thompson are the main contribution of my research.

Key-words: Multi-armed bandit, Upper Confidence Bound, Sparse UCB, non-contiguous wideband spectrum analyzer, Thompson sampling, sub-Nyquist sampling.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MAB	Multi-armed Bandit
MPMAB	Multi-play Multi armed Bandit
ADC	Analog to Digital Converter
UCB	Upper Confidence Bound
SUCB	Sparse Upper Confidence Bound
PU	Primary User
SU	Secondary User

Chapter-1

Introduction

1.1 Motivation

The goal of AIoT is facilitating networking and installation of a large count of smart devices supporting new applications such as content - based services, smart and knowledgeable surveillance networks, and automated homes, supermarkets and cities [1, 2].

Making AIoT practically available, constraints such as data processing efficiency at the perimeter and gateway, as well as lesser availability of spectrum for transmission of data among AIoT devices, must be solved. Although intelligence at the perimeter can help in minimizing data transmission expenses, effective wave band usage is required to provide trustworthy and reduced latency transmission and reception with more equipment [3-5].

Because of its extensive coverage and outstanding propagation characteristics, the sub-6 GHz frequency band is one of the most common options for AIoT networks [3, 5]. However, since the frequency band available to us is insufficient for large-scale networks and wavelength-band costs are very high, so traditional fixed frequency band spectrum distribution is not so much adequate and appropriate any more. This necessitates creative ways for identifying spectrum resources fast using dynamic spectrum allocation algorithms.

Sharing of not fixed frequency band spectrum is a strategy for next-generation wireless networks to meet reward and deployment at bigger level needs the sub-6 GHz spectrum [6]. 5G standard describes the spectrum into three categories: reserved, collaborated (2.3 GHz Europe / 3.5 GHz USA), and unreserved (2.4 GHz / 5-7 GHz / 57-71 GHz global), allowing licensed and unlicensed users to exist side by side in the shared and unlicensed bands [7].

Quasi frequency band spectrum like 700, 4400-5000, 37- 43.5, 2500-2690, 3300-4200, 5925-7125 MHz, and 24.25-29.5 and 2300-2400, GHz are also being investigated to enable diverse 5G use cases [8]. Similarly, IEEE 802.15.4 has three channel ranges for industrial internet-of-things (IIoT) networks: 1) One 250-740 MHz channel, 2) Four 3.1-4.8 GHz channels, and 3)

Eleven 6-11.6 GHz channels [9]. Spectrum characterization is difficult due to the ultra-wide and sparse nature of spectrum.

Additionally, in comparison to traditional training-based AI solutions, now we need some such adjustable and general-purpose AI solutions which can help us for meeting this uncertain atmosphere where numerous of independent networks are present.

Since we have very much frequency spectrum bands available as vacant but they are not contiguous, that is why we find some kind difficulties to access them easily, so in order to achieve this accessibility easily, there some AIoT algorithms have been made. We shall see some such algorithms which have been implemented already and also shall some novelty added in in thesis in the same direction.

1.2 Objectives and contribution

In this thesis, our aim is to design following five algorithms for selecting the subset of channels from available wide-band spectrum which are as stated below:

1. K subset learning with Upper Confidence Bound
2. K subset learning with SUCB having fixed sparsity
3. K subset learning with SUCB having variable sparsity [Proposed 1]
4. K subset learning with Thompson sampling
5. K subset learning with Sparse Thompson sampling [Proposed 2]

Third and fifth algorithms are the main contribution of our thesis, which have been given result as an excellent improvement over all the rest of the algorithms stated above in terms of time as well as complexity efficiency, that all we shall see in the upcoming chapters one by one in detail.

1.3 Organization

The thesis is organized in this manner. Chapter 2 shows the detailed literature review of many multi-armed bandit algorithms. Chapter 3 holds the detailing of our proposed work. Chapter 4 contains the results of all our work and discussion upon them. And finally in Chapter 5 conclusion and possible future scope of our work has been mentioned.

Chapter-2

Related work

Sometimes there is such situation where user has multiple number of arms to select best one among all. For handling such situation multi-armed bandit algorithms have been put forward, where user has to pull those arms and collect the rewards associated with those arms. The main aim of the user is to maximize her/his reward after certain rounds. The user is unaware of the reward behind each arm. Also, the reward associated with each arm is not of a fixed quantity. Each arm has a probability distribution associated with it. Some of them are discussed below.

2.1 Arm learning with Upper Confidence Bound

Large portion of the useful frequency spectrum was licenced to growing applications of wireless communication throughout the previous century. The fixed paradigm of spectrum distribution resulted in spectrum scarcity as the count of services that demands spectrum increased. However, a recent series of spectrum utilisation measurements [10] revealed that various spectrums were underutilised, implying that frequency band scarcity is merely fictious and because of the static allotment of the various spectrums to certain wireless services.

Dynamic Spectrum Access (DSA), which is also known as Opportunistic Spectrum Access (OSA), had been proposed like a potential result to the frequency band spectrum limiting problem.

Regarding DSA concerns, an opportunity is typically described as: a particular frequency band which is not utilized by the principal operators of that particular band spectrum at specific time in a specific geographical location [11]. Secondary user (SU), on the other hand, usually has no prior knowledge of the opportunities open to him.

The Federal Communications Commission (USA) proposed Cognitive Radio, a notion first proposed by J. Mitola [12] in 1999, as a practically available solution to this problem.

Upper Confidence Bound (UCB) algorithm is proposed as an effective in decision-making techniques for secondary users to exploit frequency band resources strategically on the basis of prior supervisions. This algorithm employs a value of index which gives an optimistic estimate for availability of resource to the secondary user.[13].

Machine learning scholars first developed policies based on UCB index computation to tackle the problem based on multi-armed bandits ([14, 15]).

Although this research is in its initial phases, still it is believed that this technique can lead to effective CAs for DSA situations. Many problems still remain, particularly when the channel cyclical occupancy behaviour does not match Bernoulli distributions or when a greater number of SUs employ the UCB-based rules to access the same core channel.

2.2 Arm learning with Sparse Upper Confidence Bound

Sparse UCB, a new algorithm came in to picture as an improvement over classical multi-armed bandit problem [16].

In classical multi-armed bandit problem, we were forced to explore each and every arm (channel) successively to get the best arm (vacant arm i.e. with highest probability of vacancy), but in case of sparse UCB we have to explore only good arms, that is why in later case the regret is proportional to the square root of total no. of arms available to us, where as in case of classical bandits, the regret was directly proportional to the total no. of arms available to us.

In the designing of Sparse UCB algorithm, there are three phases have been characterised within the same experiment:

- Round robin phase
- Force-log phase
- UCB phase

In this algorithm two lists also have been prepared:

- J-list— having active arms

- K-list— having active and sufficiently sampled arms

When the algorithm be in Round- robin phase, each and every arm is being pulled successively from the given set of arms (channels).

When the algorithm be in Force-log phase, the arms (channels) existing in the J-list are being pulled one by one.

When the algorithm be in UCB phase, the arms (channels) existing in the K-list are being pulled according to the UCB-index rule (making UCB-index value maximum should be pulled).

In the research of this Sparse UCB, it has been observed that for weak sparsity case (sparsity value is high) Sparse UCB behaves poorer to classical UCB, for moderate sparsity case (sparsity value is moderate) Sparse UCB behaves similar to classical UCB and for strong sparsity case (sparse value is less) Sparse UCB gives excellent improvements over the classical UCB.

These upper both algorithms have been implemented for learning and selecting vacant channels, but if we have to select band of channels, then for subset selection with UCB logic algorithm has been also implemented in [17], where users can get the best subset among all provided them according to UCB role.

2.3 Subset learning with Thompson sampling

Conventional multi-armed bandit algorithm is used for selecting best arm (channel) among all the given set of channels in the wireless communication network [18-23]. When we have to select a band of channels simultaneously i.e., the extension of the previous work then it is called as Multi-play Multi armed bandit (MPMAB) or Combinatorial Multi armed bandit (CMAB) [24-27], where users can select the best subset having multiple channels simultaneously.

Subset selection has been also done with Thompson sampling in [28], where users simultaneously calculate power quality index value for all the subsets with the help of beta random which is a function of rewards and regrets of all subsets given to us.

From the first iteration, users can start learning all the arms simultaneously, that is why Subset selection with Thomson gives extremely brilliant performance over both the algorithms UCB and SUCB.

There is high level of complexity associated with the Thompson learning that, in every iteration power quality index has to be calculated for all subsets available to us.

This complexity has been reduced in my novel work, which is giving us benefit also over the all algorithms stated till now. We shall see the improvements in further sections.

The performance of all algorithms is based upon the regret curve versus total count of rounds users have applied the algorithm for learning the channels or set of channels together, that regret is calculated by subtracting the calculated reward for each subset with the optimal subset at each iteration [25].

If the total channels in the selected subset is equal to total Analog to Digital converters (ADC) available with us, then the reward calculation will be in conventional way, meaning that there is no reconstruction loss, and that is why we can get the reward directly for the subset having vacant channels and no reward for subset having occupied channels [25].

But, in case if the total channels in the selected subset is greater than the total ADCs with us, then reward calculation will be in different format. If occupied channels in the selected subset are less than total ADCs with us then reward is calculated conventional way but if total number of occupied channels becomes equal or greater than the total number of ADCs available with us then users will not be entertained with any kind of reward, this is the case of reconstruction failure.

Chapter-3

Proposed work

In this chapter we will know about all the algorithms on which this project has been based. There are certain algorithms which have been already implemented but still they are being used in this project so we will have a quick glance about them briefly. And there are some new algorithms which has been implemented with novelty providing improvements and benefits over the existed algorithms, we shall know and learn about them in quite detail with their Pseudo codes.

3.1 K- subset learning using Upper Confidence Bound [17]

In the series of all the algorithms selection of optimal arm with the help of Upper confidence bound is the foremost algorithm used in my project.

So, in this algorithm player plays with all the arms randomly at the initial rounds and by doing so on gradually she learns about the nature of the arms (about their vacancy probability), and with the passage of time she settles down with the optimal one according to her knowledge and experiences.

3.1.1 The pseudo code of the K subset learning using Upper Confidence Bound

Inputs:

N= total no. of available channels

K = total no. of ADC's

SoC = set of channels [1:N]

SoS = set of subsets

L = total no. of iterations

NR= total no. of subsets

Initialization:

X_s=ones[1 X NR], % vector storing no. of times subsets have given reward.

T_s=ones[1 X NR], % vector storing no. of times subsets have been pulled.

UCB_index=zeros[1 X NR], % vector storing the upper confidence bound for each subsets

reward=zeros[1 X NR], % vector storing the reward for each subsets

mean=zeros[1 X NR], % vector storing the mean for each subsets

Outputs:

Beta (selected subset), Total no. of pulls for each subset, Regret of the algorithm

Start:

1. for 1:L
2. Select a subset from given set of all the subsets to us randomly.
3. Index of the selected subset would be chosen with the UCB rule, i.e., we have to choose that index which will make value of UCB_index maximum among all the indexes.
4. Update UCB_index ((X_s/T_s) + sqrt(alpha*log(T_s)/T_s))
5. Determine status s_beta of the channels present in beta (selected subset)
6. Determine total no. of vacant channels (s_b_c) and store their indices in beta_v
7. Update
   - X_s(n) = X_s(n) +s_b_c / K where n = index of beta,
   - reward(n) = reward(n) + p where n = index of beta and p = sum of vacancy probabilities of vacant channels presents in beta
   - T_s(n)=T_s(n) + 1 where n = index of beta,
   - mean = X_s /T_s.
8. Calculate optimal reward for best subset in similar way to calculated reward.
9. end for
10. Regret= optimal reward- calculated reward
11. Plot the cumulative sum of Regret Vs iterations

End

3.2 K- subset learning using Sparse UCB having fixed sparsity [16]

The next algorithm is learning subsets having K channels within themselves, this algorithm has come as an improvement over the earlier one. Main role of this algorithm is that, this reduces the exploration time.

We need not to explore each and every arm again and again while using this, but here we prepare two lists, first having good arms from the all arms given to us and the second list will hold the better arms filtering from the good arms existing in j-list, and finally we shall select the best optimal arm from the set of better arms according to the UCB rule.

As sparse means something which is rare i.e. not dense, specifying that in sparse UCB case we do not explore all the arms always, here we explore better arms only which are good arms. And fixed sparsity means here we shall always filter out the half of the total number of good arms available to us, then we keep on exploiting those good arms to get better and so on to get best optimal arm maximizing reward and hence minimizing regret.

This algorithm gives us very good improvement over the earlier one. The pseudo code of the SUCB having fixed sparsity is described as below.

3.2.1 Pseudo code of K subset learning algorithm using Sparse UCB having fixed sparsity

Inputs:

N= total no. of available channels

K = total no. of ADC's

SoC = set of channels [1:N]

SoS = set of subsets

L = total no. of iterations

NR= total no. of subsets

Initialization:

X_s=ones[1 X NR], % vector storing no. of times subsets have given reward.

T_s=ones[1 X NR], % vector storing no. of times subsets have been pulled.

UCB_index=zeros[1 X NR], % vector storing the upper confidence bound for each subsets

UCB_J=zeros[1 X NR], % vector storing the criteria for making list of active subsets

UCB_K=zeros[1 X NR], % vector storing the criteria for making list of active and sufficiently sampled subsets

reward=zeros[1 X NR], % vector storing the reward for each subsets

mean=zeros[1 X NR], % vector storing the mean for each subsets

j=zeros[1 X NR], % vector storing the active subsets

k= zeros[1 X NR], % vector storing the active and sufficiently sampled subsets

Phase = Round robin, % considering by default phase as round robin.

Sparsity = NR/2, % fix the sparsity value to the half of the total no. of subset

Outputs:

Beta (selected subset), Total no. of pulls for each subset, Regret of the algorithm

Start:

1. for 1:L

2. select each subset one by one for once

3. Update UCB_index, UCB_J and UCB_K

4. Update j list (according to mean (X_s / T_s) >= UCB_j)
   Update k list (according to mean (X_s / T_s) >= UCB_k)

5. If sum(j) < sparsity
   - Phase is “Round robin”
   - Select each subset one by one for once

6. Else if sum(k) < sparsity
   - Phase is “Forcelog”
   - Index for next subset to be chosen should be taken from true values in j list one by one till the algo remain in Forcelog.

7. Else
   - Phase is “UCB”
   - Index for next subset to be chosen should be taken from true values in k list making max UCB_index

8. Determine status s_beta of the channels present in beta (selected subset)

9. Determine total no. of vacant channels (s_b_c) and store their indices in beta_v

10. Update
    - X_s(n) = X_s(n) +s_b_c / K where n = index of beta,
    - reward(n) = reward(n) + p where n = index of beta and p = sum of vacancy probabilities of vacant channels presents in beta
    - T_s(n)=T_s(n) + 1 where n = index of beta,
    - mean = X_s /T_s.

11. Calculate optimal reward for best subset in similar way to calculated reward.

12. end for

13. Regret= optimal reward- calculated reward

14. Plot the cumulative sum of Regret Vs iterations

End

3.3 Proposed K subset learning with the help of Sparse UCB having varying sparsity

Since in Sparse UCB algorithm, we fixed the value of sparsity to the half of the total number of arms given to us, so in if we have much larger set given, in that case we will have even a larger set to explore, so the Sparse UCB is not much feasible in case of much larger set of arms (channels or band of channels) have been given.

So, this new algorithm, which is one of the main contributions of my thesis, provides improvement over both the algorithm described earlier (KSL with UCB and KSL with SUCB).

In this algorithm, initially we put sparsity as equal to total number of arms, then as number of rounds of playing the algorithm increase learning is being enhanced about the nature (vacant or occupied) of all arms, then we shift towards the lesser value of sparsity in order to make sure that now exploration should be done among very less, better and prone to more vacant arms.

Less value of sparsity represents that learning of the algorithm has been done quicklier than in that of higher sparsity value. If sparsity value is exactly equal to the number of arms available to us to explore, that means there is no sparsity at all, and every arm would be picked and

played successively irrespective of their vacancy and occupancy nature, that leads to a very poor performance, even poorer to UCB.

So, we move to higher values of sparsity (higher but lesser than total number of arms) which is known as weak sparsity domain, in this case also we are not moving to better arms quickly, we are forced to explore each arm unless there are at-least enough number of arms (one's equal to sparsity value) is not available in the j-list.

So again, we want to shift to some more compact value of sparsity in order to get exploration of all arms for lesser time and going to last phase may happen little earlier than the previous cases (which we do in KSL with SUCB with fixed sparsity), this gives us benefit over the classical MAB and previous two cases.

But if we want to make our exploration some more compact like a very few values of sparsity that is known as strong sparsity case, so according to our this much of research, we hope that strong sparsity case would give us excellent result over all, which does but, in some cases, in some other cases learning is not being done properly so our algorithm starts giving us selection of bad arms. This means fixing sparsity value in SUCB (apart from half of the total arms available to us) will not give much improvement over the classical MAB.

That is why we made this algorithm having all the sparsity value within the same experiment (Keeping initially sparsity high and then lowering it as learning is enhanced). This gives me proper learning of every arm and also provides outstanding performance over both the algorithms for every case, for every kind of vacancy statics, for any number of arms and for any number of channels in the subset.

Hence this a better and improved novelty in the algorithm of Sparse UCB. Pseudo code of this novel algorithm is mentioned as below.

3.3.1 Proposed pseudo code for K subset learning algorithm with Sparse UCB having varying sparsity

Inputs:

N= total no. of available channels

K = total no. of ADC's

SoC = set of channels [1:N]

SoS = set of subsets

L = total no. of iterations

NR= total no. of subsets

Initialization:

X_s=ones[1 X NR], % vector storing no. of times subsets have given reward.

T_s=ones[1 X NR], % vector storing no. of times subsets have been pulled.

UCB_index=zeros[1 X NR], % vector storing the upper confidence bound for each subsets

UCB_J=zeros[1 X NR], % vector storing the criteria for making list of active subsets

UCB_K=zeros[1 X NR], % vector storing the criteria for making list of active and sufficiently sampled subsets

reward=zeros[1 X NR], % vector storing the reward for each subsets

mean=zeros[1 X NR], % vector storing the mean for each subsets

j=zeros[1 X NR], % vector storing the active subsets

k= zeros[1 X NR], % vector storing the active and sufficiently sampled subsets

Phase = Round robin, % considering by default phase as round robin.

Sparsity = NR, % initially the sparsity value equal to the total no. of subset

Outputs:

Beta (selected subset), Total no. of pulls for each subset, Regret of the algorithm

Start:

1. for 1:L

2. select each subset one by one for once

3. Update UCB_index, UCB_J and UCB_K

4. Update j list (according to mean (X_s / T_s) >= UCB_j)
   Update k list (according to mean (X_s / T_s) >= UCB_k)

5. If sum(j) < sparsity
   - Phase is “Round robin”

- Select each subset one by one for once
- Counter=0
- Counter new=0

6. Else if sum(k) < sparsity
   - Phase is “Forcelog”
   - Index for next subset to be chosen should be taken from true values in j list one by one till the algo remain in Forcelog.
   - Counter new=0
   - Counter= Counter+1
   - If Counter > N and sparsity > 4
   - Sparsity=sparsity/2

7. Else
   - Phase is “UCB”
   - Index for next subset to be chosen should be taken from true values in k list making max UCB_index
   - Counter new= Counter new+1
   - If Counter new > NR and sparsity > 4
   - Sparsity=sparsity/2

8. Determine status s_beta of the channels present in beta (selected subset)

9. Determine total no. of vacant channels (s_b_c) and store their indices in beta_v

10. Update
    - X_s(n) = X_s(n) +s_b_c / K where n = index of beta,
    - reward(n) = reward(n) + p where n = index of beta and p = sum of vacancy probabilities of vacant channels presents in beta
    - T_s(n)=T_s(n) + 1 where n = index of beta,
    - mean = X_s /T_s.

11. Calculate optimal reward for best subset in similar way to calculated reward.

12. end for

13. Regret= optimal reward- calculated reward

14. Plot the cumulative sum of Regret Vs iterations

End

3.4 K subset learning with the help of Thompson sampling

This algorithm is very simple and very basic but it gives even more excellent results than the all three algorithms described till now.

In this, user has to calculate power quality index value for each subset at each iteration with the help of rewards and regret calculated for each arm at each iteration. Then the index where power quality index value is maximum that is chosen as the index of next subset to be selected.

There is no chance of poor learning in this algorithm because it starts calculating the power quality index of each arm from the first iteration and it calculates this for each arm and every iteration, so learning is being very properly in this algorithm.

Also, it learns the best arm very quickly so it is very much time efficient, so probability of getting reward is quite higher that leads to quite lower regret compare to all the upper three algorithms.

Pseudo code of K subset learning with Thompson is as follows.

3.4.1 Pseudo code for K subset learning algorithm with Thompson sampling [28]

Inputs:

N= total no. of available channels

K = total no. of ADC's

SoC = set of channels [1:N]

SoS = set of subsets

L = total no. of iterations

NR= total no. of subsets

Initialization:

X_s=ones[1 X NR], % vector storing no. of times channels have given reward.

T_s=ones[1 X NR], % vector storing no. of times subsets have been pulled.

reward=zeros[1 X NR], % vector storing the reward for each subsets

Outputs:

Beta (selected subset), Total no. of pulls for each subset, Regret of the algorithm

Start:

1. for 1:L

2. Select the best subset from all the subsets available to us for once according to Thompson rule.
    - By calculating Power quality index (Q_s) for all subsets
    - Q_s = Beta (X_s, T_s – X_s)
    - Index of selected subset = maximum (Q_s)

3. Determine status s_beta of the channels present in beta (selected subset)

4. Determine total no. of vacant channels (s_b_c) and store their indices in beta_v

5. Update
    - X_s(n) = X_s(n) +s_b_c / K where n = index of beta,
    - reward(n) = reward(n) + p, where n = index of beta and p = sum of vacancy probabilities of vacant channels presents in beta
    - T_s(n)=T_s(n) + 1 where n = index of beta,

6. Calculate optimal reward for best subset in similar way to calculated reward.

7. end for

8. Regret= optimal reward- calculated reward

9. Plot the cumulative sum of Regret Vs iterations

End

3.5 Proposed K subset learning algorithm with the help of Sparse Thompson sampling

Since KSL with Thompson was showing outstanding performance over the mentioned algorithms as we have seen section 3.4, but still, it also has a disadvantage and that is complexity is very high in this algorithm.

If there is much larger number of arms given to user to find out the best one among all of them, then she has to calculate power quality index for all arms, which leads to quite higher complexity, even if algorithm identifies the best one after certain rounds of exploration still power quality index is keep on being calculated for each iteration.

So, to reduce this complexity I have designed a new algorithm, which is the second novelty of my thesis. Here I have introduced varying sparsity concept with the Thompson.

There are three phases of this algorithm:

1. Thompson Phase
2. Restricted Thompson Phase
3. Most Restricted Thompson Phase

Here again two lists are being prepared:

i. J-list→ list having the active arms
ii. K-list→ list having the active and sufficiently sampled arms

So, when learning of arms will be started, simultaneously we prepare these two lists, and by the time first list will have as much number of true values equal to the sparsity defined, we keep on calculating power quality index value for each and every arm given to us.

Once J-list will cross the limits of true values than the sparsity, now power quality index values will be calculated only for those arms, which indexes are having true in J-list, and optimal arm would be coming from that list only.

And again, now we shall check the K-list, if once this crosses the limits of true values than the sparsity, now for selecting best subset we shall calculate power quality index value only for those arms, which indexes are having true in K-list.

That is how we became able to reduce the complexity of K subset learning with Thompson, which was the drawback of this algorithm. Varying sparsity helps to learn all the arms properly in the beginning and later on reducing sparsity leads us towards the best arms only with reduced complexity.

And if we observe the performance of this algorithm, it is giving super excellent performance over all the four algorithms described in previous sections of this chapter in all the scenario in terms of different vacancy probability statistics, any number of channels present in the selected subset, any number of total subsets given to us to explore and of course the least complexity.

Pseudo code of this novel algorithm is described below.

3.5.1 Proposed pseudo code for K subset learning algorithm with the help of Sparse Thompson sampling

Inputs:

N= total no. of available channels

K = total no. of ADC's

SoC = set of channels [1:N]

SoS = set of subsets

L = total no. of iterations

NR= total no. of subsets

Initialization:

X_s=ones[1 X NR], % vector storing no. of times subsets have given reward.

T_s=ones[1 X NR], % vector storing no. of times subsets have been pulled.

UCB_J=zeros[1 X NR], % vector storing the criteria for making list of active subsets

UCB_K=zeros[1 X NR], % vector storing the criteria for making list of active and sufficiently sampled subsets

reward=zeros[1 X NR], % vector storing the reward for each subsets

mean=zeros[1 X NR], % vector storing the mean for each subsets

j=zeros[1 X NR], % vector storing the active subsets

k= zeros[1 X NR], % vector storing the active and sufficiently sampled subsets

Phase = Thompson, % considering by default phase as round robin.

Sparsity = NR/2, % initially the sparsity value equal to the total no. of subset

Outputs:

Beta (selected subset), Total no. of pulls for each subset, Regret of the algorithm

Start:

1. for 1:L

2. Select the best subset from all the subsets available to us for once according to Thompson rule.

    - By calculating Power quality index (Q_s) for all subsets
    - Q_s = Beta (X_s, T_s – X_s)
    - Index of selected subset = maximum (Q_s)

3. Update UCB_J and UCB_K

4. Update j list (according to mean (X_s / T_s) >= UCB_j)
   Update k list (according to mean (X_s / T_s) >= UCB_k)

5. If sum(j) < sparsity
    - Phase is “Thompson”
    - Select the best subset from all the subsets available to us for once according to Thompson rule.
    - Counter=0
    - Counter new=0

6. Else if sum(k) < sparsity
    - Phase is “Restricted Thompson”
    - Select the best subset i.e. index of best subset would be taken from all the true values available in the J-list according to Thompson rule.
    - Counter new=0
    - Counter= Counter+1
    - If (Counter > N) and (sparsity > 4)
    - Sparsity=sparsity/2

7. Else
    - Phase is “Strictly Restricted Thompson”
    - index of best subset would be chosen from all the true values available in k list according to Thompson rule.
    - Counter new= Counter new+1
    - If Counter new > NR and sparsity > 4
    - Sparsity=sparsity/2

8. Determine status s_beta of the channels present in beta (selected subset)

9. Determine total no. of vacant channels (s_b_c) and store their indices in beta_v

10. Update
    - X_s(n) = X_s(n) +s_b_c / K where n = index of beta,
    - reward(n) = reward(n) + p, where n = index of beta and p = sum of vacancy probabilities of vacant channels presents in beta
    - T_s(n)=T_s(n) + 1 where n = index of beta,
    - mean = X_s /T_s.

11. Calculate optimal reward for best subset in similar way to calculated reward.

12. end for

13. Regret= optimal reward- calculated reward

14. Plot the cumulative sum of Regret Vs iterations

End

3.6 Regret Calculation

For this project we have calculated regret by using following methodology. Calculate the optimal reward directly as the summation of vacancy probabilities of the channels present in the best subset. Then calculate regret by subtracting the addition of vacancy probabilities of channels present in selected subset from the optimal reward.

3.7 Summary

Till this chapter we have quite sound knowledge about performance and behavior of all the five algorithms theoretically. In the next chapter we will see these things practically through simulation and discuss the characteristics in detail.

Chapter-4

Results and Discussion

4.1 Simulation setup

All the five algorithms have been simulated and verified on MATLAB.

4.2 Simulation Parameters

All the algorithms are simulated over the average of ten experiments and for ten thousand rounds for the following parameters.

Table 4.1: Simulation parameters for different scenarios

Scenario count	Total channels (N)	Total channels in the subset (K)	Vacancy Probability Statistics of channels (P)	Total arms to explore (NR)
Scenario 1	8	2	0.95:0.05:0.6	28
Scenario 2	8	3	0.95:0.05:0.6	56
Scenario 3	12	2	0.95:0.05:0.4	66
Scenario 4	12	3	0.95:0.05:0.4	220
Scenario 5	8	2	0.9 0.4 0.7 0.8 0.5 0.1 0.2 0.3	28
Scenario 6	8	3	0.9 0.4 0.7 0.8 0.5 0.1 0.2 0.3	56

4.3 Simulation Results

4.3.1 Scenario 1

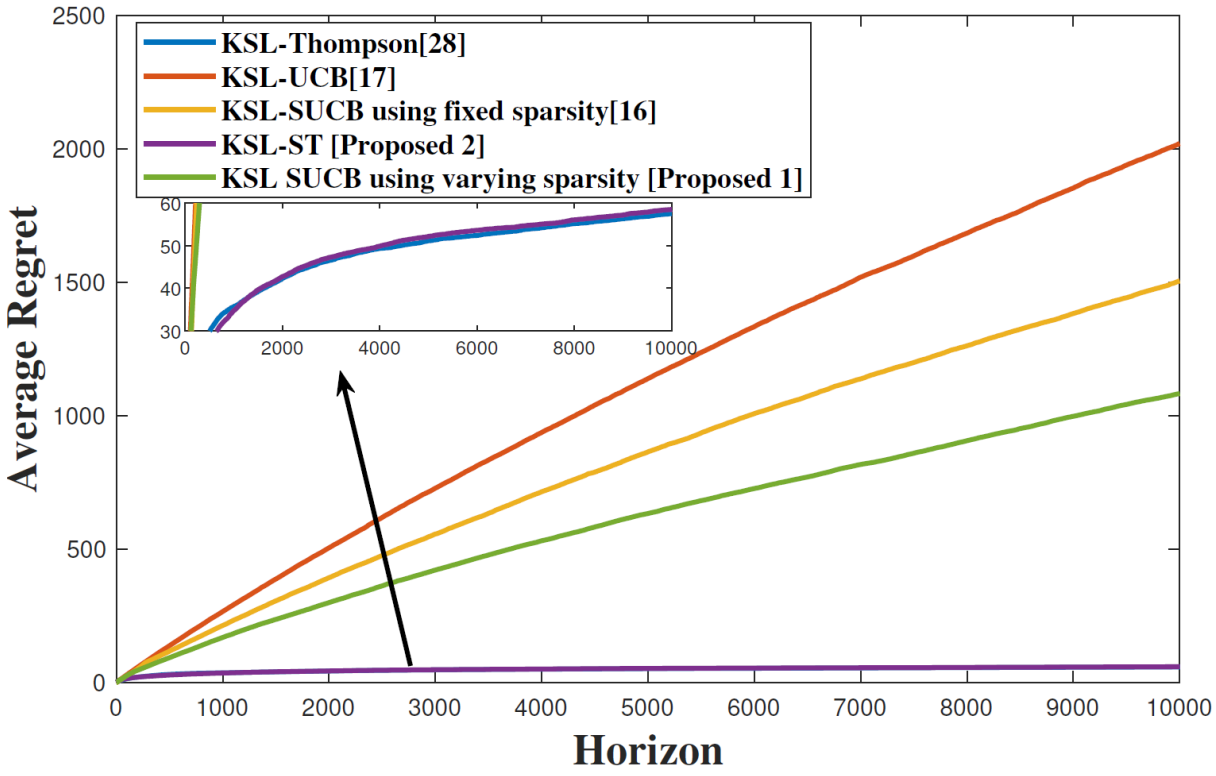

Fig 4.1: The plots showing the comparison of average regret of the existing state-of-the-art and the proposed algorithms for scenario 1: N= 8, K=2, Prob = [0.9:0.05:0.6]

Figure 4.1 presents the comparison among regret plots among all the algorithms which we have gone through in detailing in chapter 3. Here graphs have been plotted for both kind of regret calculation. In this figure, the parameters are according to case 1, where total no. of subsets are 28 and statistics are more prone to be vacant. That is why regret value is lesser.

The total number of times each arm has been pulled according to the algorithms UCB, SUCB with fixed sparsity, SUCB with varying sparsity, Thompson and Sparse Thompson respectively for the parameters of this scenario. Since here vacancy probability statistics are such as that first subset holds the maximum chance to be vacant compare to others, but there are others also with very lesser difference than the first one.

So, if we observe carefully then we would find that in all the algorithms, first subset is being pulled maximum number of times compare to others but there are some differences while selecting the best in different algorithms.

In UCB, of course compare to other arms the first one is being pulled maximum times, but others are also being pulled sufficient number of times, coming in to SUCB with fixed sparsity we can see that still half of the arms are being pulled quite sufficient number of times along with the first one maximum times, and then if we observe SUCB with varying sparsity, in this the best one is being selected quite number of times, but sometimes second best can also be pulled maximum no. of times, this because of statistics and number of rounds player plays the algorithm. But even if some times user selects the second-best arm, then also SUCB with fixed sparsity is improvement over UCB and SUCB with varying sparsity is improvement over SUCB with fixed sparsity, this is because the latter two algorithms do not entertain bad arms maximum number of times. Coming to KSL with Thompson, it selects the best one maximum number of times along with quite a greater difference than the other all arms, and some arms having least vacancy probabilities among all are not even being pulled a single time. The only problem with this was complexity, that is being reduced by using the last algorithm, KSL-Sparse Thompson. In KSL-Sparse Thompson the best arm is pulled maximum number of times with a very large difference than the others along with maintaining the lesser complexity.

4.3.2 Scenario 2

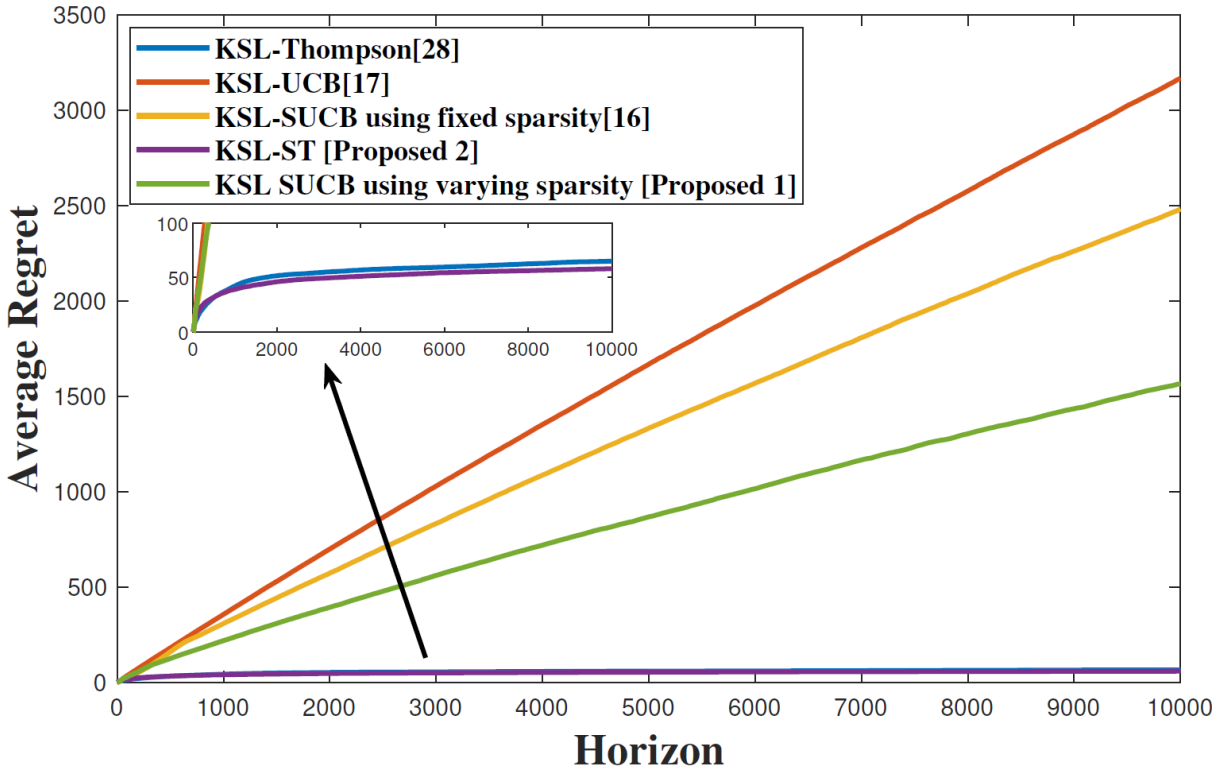

Fig 4.2: The plots showing the comparison of average regret of the existing state-of-the-art and the proposed algorithms for scenario 1: N= 8, K=3, Prob = [0.9:0.05:0.6]

Figure 4.2 also shows the comparison among regret plots among all the algorithms which we have gone through in detailing in chapter 3. In this figure, the parameters are according to case 2, where total no. of subsets are 56 and statistics are more prone to be vacant (same as the previous case). In this case total arms to be explored are greater than the previous case that is why regret value is higher than the previous one.

The total number of times each arm has been pulled according to the algorithms UCB, SUCB with fixed sparsity, SUCB with varying sparsity, Thompson and Sparse Thompson respectively for the parameters of this case. Since here vacancy probability statistics are such as that first subset holds the maximum chance to be vacant compare to others. So, selection of arms has been done accordingly for all the algorithms as it has been discussed in detail in section 4.4.1. Since statistics are such that user gets confused sometimes and learns the second best in case of SUCB (for both types of sparsity), which is very much practically possible situation.

4.3.3 Scenario 3

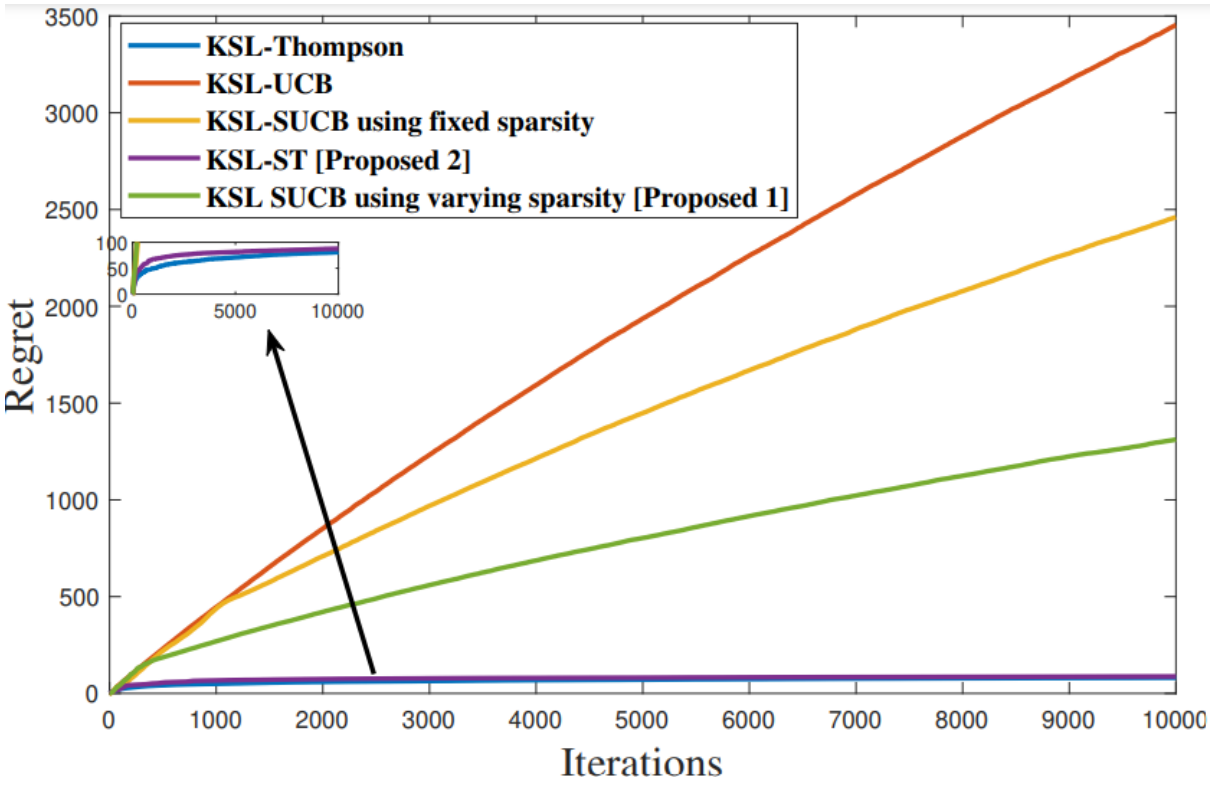

Fig 4.3: The plots showing the comparison of average regret of the existing state-of-the-art and the proposed algorithms for scenario 1: N= 12, K=2, Prob = [0.9:0.05:0.4]

Fig 4.3 again shows the comparison among regret plots among all the algorithms which we have gone through in detailing in chapter 3. In this figure, the parameters are according to case 3, where total no. of subsets are 66 and statistics are kind of both types vacant as well as occupied (not same as the previous case). In this case total arms to be explored are again greater than the previous both cases that is why regret value is higher in this than the previous both cases.

Total number of times each arm has been pulled according to the algorithms UCB, SUCB with fixed sparsity, SUCB with varying sparsity, Thompson and Sparse Thompson respectively for the parameters of this case. Here also learning of all the arms in every algorithm is as per the logic.

4.3.4 Scenario 4

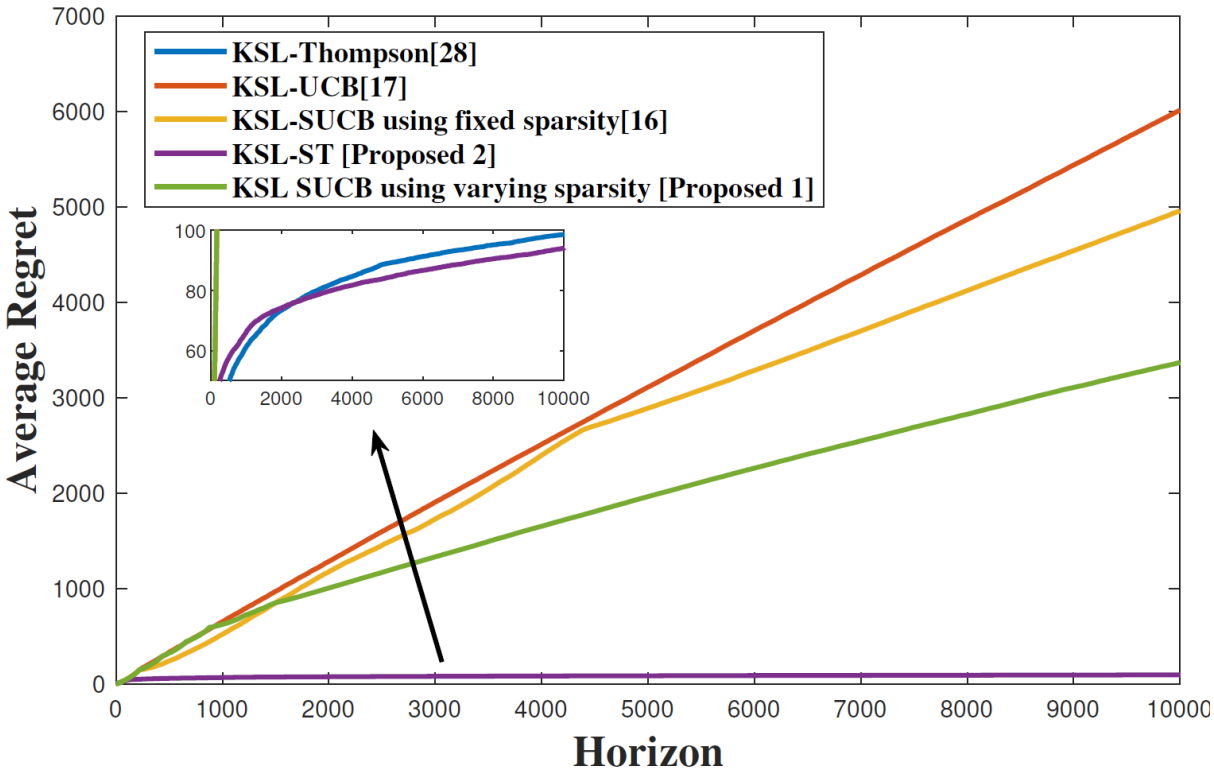

Fig 4.4: The plots showing the comparison of average regret of the existing state-of-the-art and the proposed algorithms for scenario 1: N= 12, K=3, Prob = [0.9:0.05:0.4]

Figure 4.4 shows the comparison among regret plots among all the algorithms which we have gone through already in detailing in the previous chapter. In this figure, the parameters are according to case 4, where total no. of subsets are 220 and statistics are kind of both types vacant as well as occupied (same as the previous case). In this case total arms to be explored are much greater than the previous all cases that is why regret value is higher in this than the previous all cases.

The total number of times each arm has been pulled according to the algorithms UCB, SUCB with fixed sparsity, SUCB with varying sparsity, Thompson and Sparse Thompson respectively for the parameters of this case. Here also learning of all the arms in every algorithm is as per the logic.

4.3.5 Scenario 5

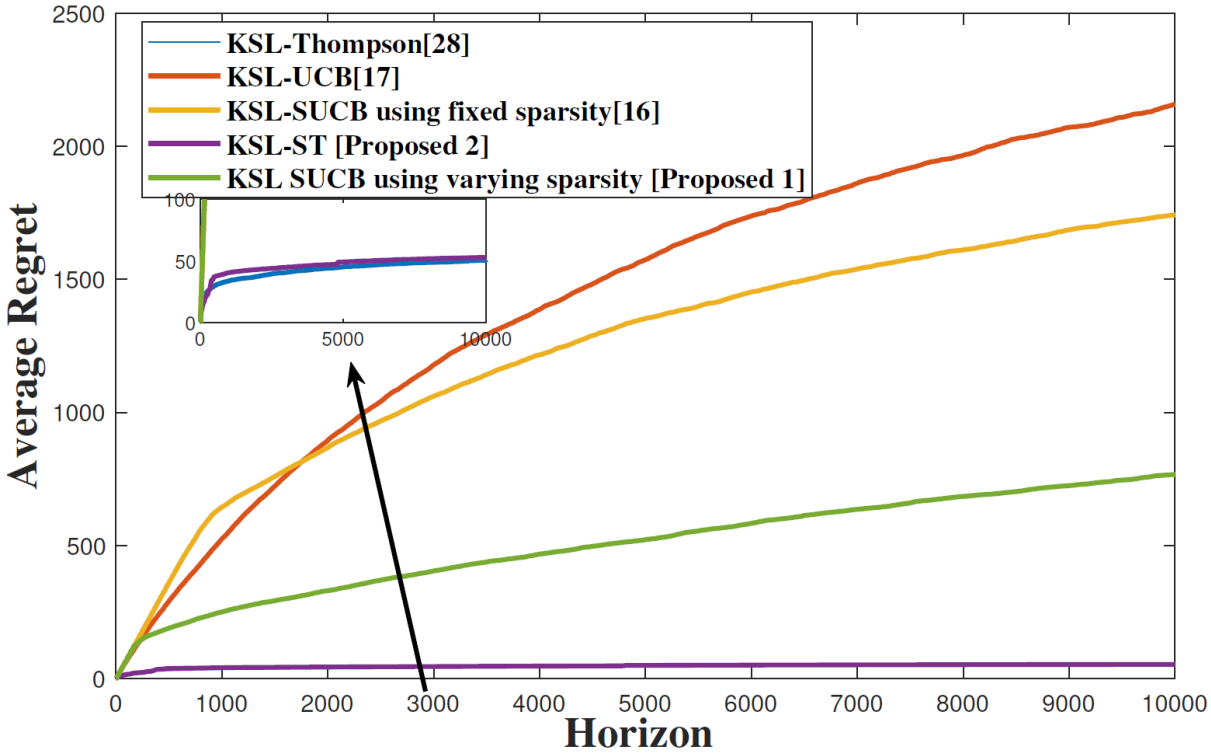

Fig 4.5: The plots showing the comparison of average regret of the existing state-of-the-art and the proposed algorithms for scenario 1: N= 8, K=2, Prob= [0.9 0.4 0.7 0.8 0.5 0.1 0.2 0.3]

Figure 4.5 also presents the comparison among regret plots among all the algorithms which we have gone through already in detailing in the previous chapter. In this figure, the parameters are according to case 5, where total no. of subsets are 28 and statistics are exactly both types vacant as well as occupied (0.1 - 0.9 in random order, just to check the authenticity of the algorithm). In this case total arms to be explored are lesser than the previous all cases except the first one, that is why regret value is lesser in this than the previous all cases and quite similar to the first one, because the best subset for both the cases first and fifth are having quite similar vacancy probability.

The total number of times each arm has been pulled according to the algorithms UCB, SUCB with fixed sparsity, SUCB with varying sparsity, Thompson and Sparse Thompson respectively for the parameters of this case. Here also learning of all the arms in every algorithm is as per the logic.

4.3.6 Scenario 6

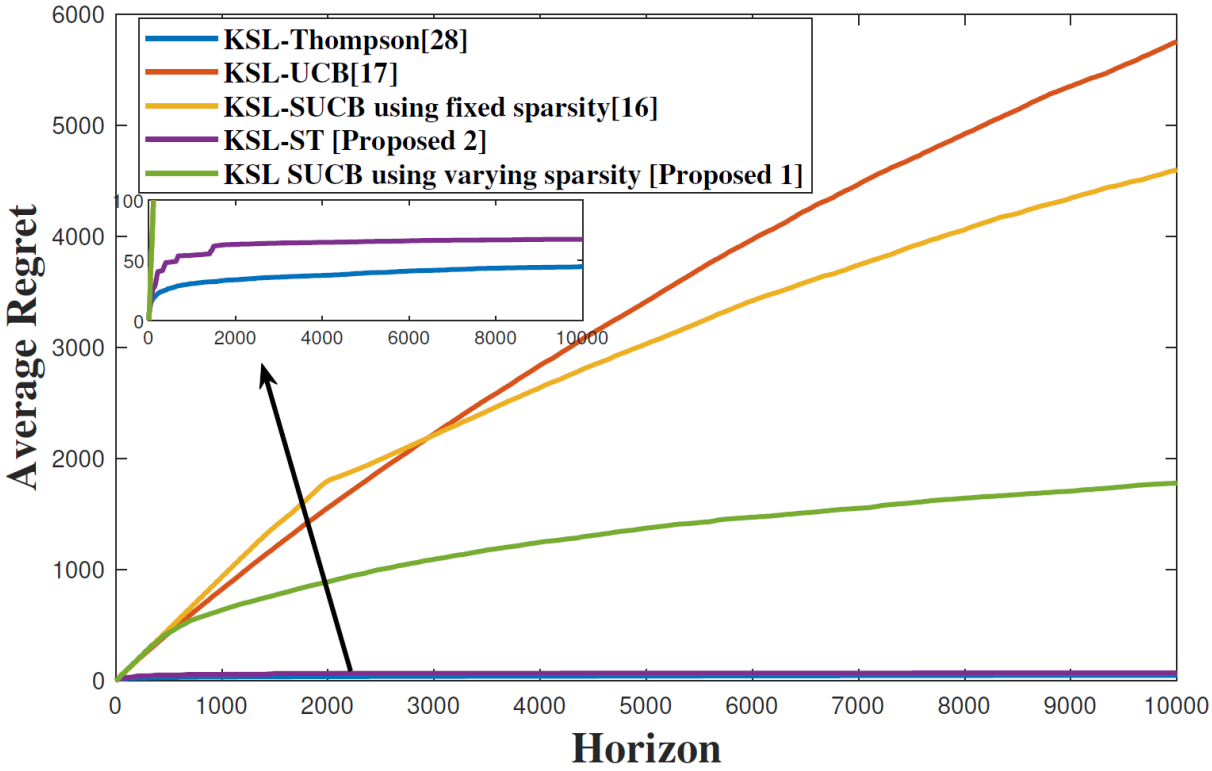

Fig 4.6: The plots showing the comparison of average regret of the existing state-of-the-art and the proposed algorithms for scenario 1: N= 8, K=3, Prob= [0.9 0.4 0.7 0.8 0.5 0.1 0.2 0.3]

Figure 4.6 again presents the comparison among regret plots among all the algorithms which we have gone through already in detailing in the previous chapter. In this figure, the parameters are according to case 6, where total no. of subsets are again 56 and statistics are exactly both types vacant as well as occupied (0.1 - 0.9 in random order, just to check the authenticity of the algorithm i.e., same as just previous case). In this case total arms to be explored are higher than the previous cases (only first and fifth), equal to the second case, and lower than the third case), that is why regret value is in the similar order with all cases.

The total number of times each arm has been pulled according to the algorithms UCB, SUCB with fixed sparsity, SUCB with varying sparsity, Thompson and Sparse Thompson respectively for the parameters of this case. Here also learning of all the arms in every algorithm is as per the logic.

4.4.7 Summary

Discussion till now is about the differences among all the cases, now if we talk about difference among regret of all the algorithms for all cases, so we find that value of regrets are in descending order for the algorithms KSL-UCB, KSL-SUCB with fixed sparsity and KSL-SUCB with varying sparsity respectively, for KSL-Thompson and KSL-SThompson regret is quite similar to each other and lowest among all the algorithms for all the cases.

This is the exactly same behavior which we were analyzing in the chapter 3 theoretically, here we are verifying the same behavior practically by simulation.

We have plotted comparison of regret graphs for all cases among all algorithms for two different types of regret calculation. Regret plot of the upper part of the result is with 1^{st} methodology of regret calculation and lower part is with the 2^{nd} methodology of regret calculation. If we observe carefully then we will be able to see that value of the regret in first part is always being remain higher than in the second part.

It is because in the first regret calculation, we are considering the situation for the channel existing in the subset to be vacant or occupied but in the second regret calculation we are directly considering the subset to be fully vacant. This is the reason regret in the first one is being remain higher than in the second one.

Chapter-5

Conclusion and Future scope

5.1 Conclusion

In this thesis basically we are dealing with subsets having K channels within themselves. K is the total amount of ADCs with us, so the same number of channels in the subsets represents that there is no chance of reconstruction failure. We shall always get the channels providing us reward if they would be vacant.

So, for selecting optimal subset from all the subsets given to us, several algorithms have been designed and existed already. In the same context we have also designed two novel algorithms, which are providing excellent improvement over the existing algorithms in all aspects like for all kind of vacancy probability statistics, for any count of channels and ADCs with reduced complexity.

In the first algorithm, we select the best subset having K channels within it from the various subsets available to us with the help of UCB method, there we select each and every subset available to us and then select the subset making the upper confidence bound maximum among all.

In the second algorithm, we select the best subset having K channels within it from the various subsets available to us with the help of Sparse UCB method, where we able to explore and exploit the better subsets among all the subsets and then finally select the best one from the better subsets.

In the third algorithm, initially we explore each and every subset given to us then we start exploiting the subsets, thus we keep to learn the behavior of the subset and with the passage of time we keep reducing exploration of each subset, means we focus towards the better subsets and finally we get the best subsets efficiently.

In the fourth algorithm, we learn all the subsets provided to us according to Thompson sampling where we can identify the best subset among all very quickly and in the very efficient way.

The fifth algorithm is the most efficient algorithm among all because of the following reasons:

- It provides us the best subset very quickly as the fourth one.
- Also, the complexity is extremely lesser in this algo, because we keep reducing the no. of subsets to explore with Thompson sampling.
- Lesser complexity and quickly identification of the best subset makes this algorithm most efficient and useful to users.

5.2 Future scope

With many different types of algorithms for no reconstruction failure case, we have seen in this thesis. Now if we have such subsets which have greater number of channels within themselves than the total count of ADCs available, then there is chance of occurring reconstruction failure.

Since greater number of channels within the subsets than total ADCs available obviously leads to reconstruction failure some times, but it is always beneficial to select such band or subset which include larger number of channels within themselves over the subsets having lesser channels within themselves.

So, there are some following directions of future scope for our thesis:

- Extension of the proposed algorithms for the selection of optimal subset having larger and vacant channels within itself from the given set of all subsets having different count of channels within themselves, this situation will be actually real time scenario of shared frequency band spectrum.
- Design of a reconfigurable architecture where users can switch dynamically between arms and algorithms.

References

[1] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu and X. Du, "From IoT to 5G I-IoT: The Next Generation IoT-Based Intelligent Algorithms and 5G Technologies," in IEEE Communications Magazine, vol. 56, no. 10, pp. 114-120, Oct. 2018.

[2] J. Zhou, Y. Wang, K. Ota and M. Dong, "AAIoT: Accelerating Artificial Intelligence in IoT Systems," in IEEE Wireless Communications Letters, vol. 8, no. 3, pp. 825-828, June 2019.

[3] L. Li and A. Ghasemi, "IoT-Enabled Machine Learning for an Algorithmic Spectrum Decision Process," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1911-1919, April 2019.

[4] H. Song, J. Bai, Y. Yi, J. Wu and L. Liu, "Artificial Intelligence Enabled Internet of Things: Network Architecture and Spectrum Access," in IEEE Computational Intelligence Magazine, vol. 15, no. 1, pp. 44-51, Feb. 2020.

[5] W. Yao, F. Khan, M.A. Jan, et al.,"Artificial intelligence-based load optimization in cognitive Internet of Things," in Springer: Neural Computing & Applications vol. 32, pp. 16179–16189, March 2020.

[6] W. S. H. M. W. Ahmad et al., "5G Technology: Towards Dynamic Spectrum Sharing Using Cognitive Radio Networks," in IEEE Access, vol. 8, pp. 14460-14488, Jan. 2020.

[7] S. Parkvall, E. Dahlman, A. Furuskar and M. Frenne, "NR: The New 5G Radio Access Technology," in IEEE Communications Standards Magazine, vol. 1, no. 4, pp. 24-30, Dec. 2017.

[8] Y. Kim et al., "New Radio (NR) and its Evolution toward 5GAdvanced," in IEEE Wireless Communications, vol. 26, no. 3, pp. 2-7, June 2019.

[9] IEEE, "802.15.4a-2007 - IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks - Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," 2007.

[10] Federal Communications Commission. "Spectrum policy task force report." November 2002.

[11] P. Kolodzy and al. "Next generation communications: Kickoff meeting." In Proc. DARPA, October 2001.

[12] J. Mitola and G.Q. Maguire, "Cognitive radio: making software radios more personal." Personal Communications, IEEE, 6:13–18, August 1999.

[13] Wassim Jouini, Damien Ernst, Christophe Moy, Jacques Palicot, "Upper confidence bound based decision-making strategies and dynamic spectrum access."

[14] T.L. Lai and H. Robbins. "Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics", 6:4–22, 1985.

[15] R. Agrawal, "Sample mean-based index policies with o(log(n)) regret for the multi-armed bandit problem." Advances in Applied Probability, 27:1054–1078, 1995.

[16] Joon Kwon, Vianney Perchet, Claire Vernade*, “Sparse Stochastic Bandits”, June 6, 2017.

[17] W. Chen, Y. Wang, and Y. Yuan, “Combinatorial multi-armed bandit: General framework and applications,” inInternational Conference on Machine Learning (ICML), vol. 28, no. 1, pp. 151–159, June 2013, Georgia, USA.

[18] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. -C. Chen and L. Hanzo, “Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks,” in IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1472-1514, Jan. 2020.

[19] S. Takeuchi, M. Hasegawa, K. Kanno, et al., “Dynamic channel selection in wireless communications via a multi-armed bandit algorithm using laser chaos time series,” in Scientific Report, vol. 10, no. 1574, Jan. 2020.

[20] F. Li, D. Yu, H. Yang, J. Yu, H. Karl and X. Cheng, “Multi-Armed-Bandit-Based Spectrum Scheduling Algorithms in Wireless Networks: A Survey,” in IEEE Wireless Communications, vol. 27, no. 1, pp. 24-30, February 2020.

[21] J. Zhang, Z. Li and S. Tang, “Value of Information Aware Opportunistic Duty Cycling in Solar Harvesting Sensor Networks,” in IEEE Trans on Industrial Informatics, vol. 12, no. 1, pp. 348-360, Feb. 2016.

[22] A. Castiglione, G. Cozzolino, F. Moscato and V. Moscato, “Cognitive Analysis in Social Networks for Viral Marketing,” in IEEE Trans on Industrial Informatics, Sept 2020.

[23] S. J. Darak and M. Hanawal, “Multi-player Multi-armed Bandits for Stable Allocation in Heterogeneous Ad-Hoc Networks,” in IEEE JSAC Special Issue on Machine Learning in Wireless Communications, vol.37, no. 10, pp. 2350-2363, Oct. 2019.

[24] W. Chen, Y. Wang, and Y. Yuan, “Combinatorial multi-armed bandit: General framework and applications,” inInternational Conference on Machine Learning (ICML), vol. 28, no. 1, pp. 151–159, June 2013, Georgia, USA.

[25] J. Komiyama, J. Honda and H. Nakagawa, “Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays,” arXiv preprint arXiv:1506.00779, 2015.

[26] W. Chen, Y. Wang, Y. Yuan, Q. and Wang, “Combinatorial multi-armed bandit and its extension to probabilistically triggered arms,” in Journal of Machine Learning Research, vol. 17, no. 50, pp. 1–33, April 2016.

[27] S. Wang and W. Chen. “Thompson Sampling for Combinatorial Semi- Bandits,” in International Conference on Machine Learning (ICML), vol. 80, no. 1, pp. 5114-5122, July 2018, Stockholm, Sweden.

[28] Himani Joshi, Shubhrajit Santra, Sumit J. Darak, Manjesh K. Hanawal and S. V. Sai Santosh, “Multi-Play Multi-Armed Bandit Algorithm Based Sensing of Non-Contiguous Wideband Spectrum for AIoT Networks”.