Joint reconfigurable intelligent surface selection and Frequency allocation in 5G cellular network

A Dissertation Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Master of Technology
in
Computer Science

by

Dipika Sarjerao Shende
(Roll No. CS2130)

Under the supervision of
Prof. Sasthi C. Ghosh

INDIAN STATISTICAL INSTITUTE

June 2023

INDIAN
STATISTICAL
INSTITUTE

Dipika Sarjerao Shende

Indian Statistical Institute

203 Barrackpore Trunk Road

Kolkata 700108, India

E-mail:dipikashende08@gmail.com

DECLARATION

I, **Dipika Sarjerao Shende (Roll No: CS2130)**, hereby declare that, this report entitled **"Joint reconfigurable intelligent surface selection and Frequency allocation in 5G cellular network"** submitted to Indian Statistical Institute towards partial requirement of **Master of Technology** in **Computer science**, is an original work carried out by me under the supervision of Sasthi C. Ghosh and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Dipika Sarjerao Shende

Indian Statistical Institute

June, 2021.

INDIAN
STATISTICAL
INSTITUTE

Professor Sasthi C. Ghosh

Indian Statistical Institute
203 Barrackpore Trunk Road
Kolkata 700108, India
E-mail:mobile.wifi@gmail.com

CERTIFICATE

This is to certify that the work contained in this project report entitled "**Joint reconfigurable intelligent surface selection and Frequency allocation in 5G cellular network**" submitted by **Dipika Sarjerao Shende** (Roll No. **CS2130**) to the Indian Statistical Institute towards the partial requirement of **Master of Technology** in **Computer Science** has been carried out by her under my supervision and that it has not been submitted elsewhere for the award of any degree.

Indian Statistical Institute
Date: June, 2023.

Dr. Sasthi C. Ghosh
Professor, Advanced Computing
and Microelectronics Unit,
Indian Statistical Institute, Kolkata
Thesis Supervisor.

Acknowledgements

I would like to express my heartfelt gratitude to my advisor Prof. Sasthi C. Ghosh for guiding me along this journey through his continous support and motivation. I thoroughly enjoyed discussion with him, which were very helpful in improving my knowledge as well as in offering an insight into the subject. I indeed feel very lucky to have got an advisor like him, who also helped me cope with troubles in my life.

Even a walk on toilsome path is a cake walk, if you have company of good friends. I take this opportunity to thank my friends who made my stay in Kolkata joyous and delightful through endless discussions about computer science and in general.

I thank my thesis committee and the referees for their efforts.

I gratefully acknowledge support from the Indian Statistical Institute.

Dipika Sarjerao Shende
ISI, 16 June 2023.

Dedicated to my Parents

Abstract

In this thesis, we are jointly dealing reconfigurable intelligent surface (RIS) selection and frequency allocation for device to device(D2D) communication. RIS is a two dimensional surface such that it can reflect the signal which is incident on it to the desire direction. So RISs can be used to bypass the obstacles hence it can be effectively used in D2D communication. We aim to design a joint RIS and channel allocation selection mechanism for device to device (D2D) communication in 5G cellular network such that 1) maximum number of requesting pairs are served, 2) there is no interference between any requesting pairs, and 3) at most k channels are used. Suppose n D2D pairs are requesting and there are m RISs already deployed. The interference relationship between the requesting pairs at a particular time instant t can be modelled as an interference graph $G(t)$. In $G(t)$, each requesting pair with RISs around that pair makes clique which represents a vertex and there is an edge between two requesting pairs if they interfere each other. It is important to note that $G(t)$ explicitly depends on the selection of RISs. Moreover, for a given $G(t)$, number of requesting pairs can be served explicitly depends on the channel allocation mechanism. We aim to select the RISs and also find a channel allocation mechanism such that the maximum number of requesting pairs can be served with k available channels.

Contents

Chapter 1

Introduction

In device-to-device communications [1], there are many problems. Between two devices, there may be obstacles or not. Here obstacle any object for example building, trees, house, etc. [2, 3] So sometimes there is direct communication is possible between two users but sometimes there are obstacles between two users. Direct communicating pair means there is no obstacle between the users of that communicating pair. Consider the following diagram in which user A and user B is direct communicating pair which is shown in Figure 1 and user C and user D is the communicating pair in which obstacle is present which is shown in Figure 2.

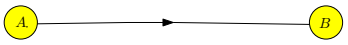


Figure 1

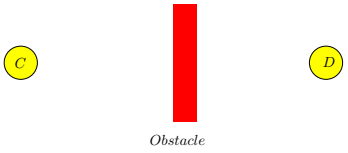
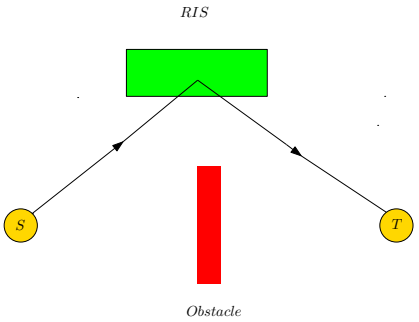


Figure 2

So for the direct communicating pair A and B, after giving frequency communication will be happen but for the communicating pair C and D communication will not be happen after giving frequency because there is obstacle between C and D. So in this situation RIS will be used [4]. First we see how RIS bypass the obstacle. Consider the following diagram,



Suppose user S and user T wants to communicate and there is obstacle between them. Suppose RIS is available near that pair. Then when S transmits the signal to RIS and when this signal incident on RIS then RIS reflects this signal and this signal reaches to user T. So in that way, RIS bypass the obstacle between the users and communication is possible between users S and T.
So for the above pair of users C and D, if there is RIS available near that pair then communication will be possible which is shown in following figure,

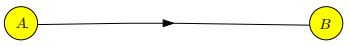


Figure 1

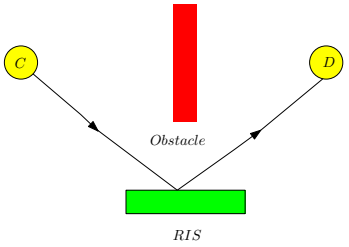


Figure 3

We see that if there is obstacle present between pair then using RIS this obstacle bypass and communication possible.But suppose there are two or three pairs of users want to communicate and RISs are available around that pairs then question is which RIS among these available RISs should be select for any of pair.Consider the following scenario,

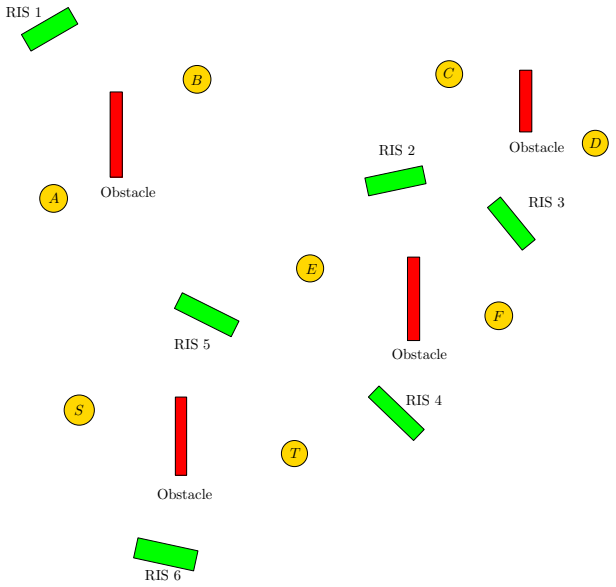


Figure 4

In this figure 4, 3 RISs are available for user E and user F, 2 RISs are available for user S and user T and 2 RISs are available for user C and user D. Which RIS will be choose for which pair is a question.

Also to activate the communication between pairs after choosing RIS, proper frequency should be assign to that pair [5]. But another question arises that which frequency should be given to pair. Also allocating frequency to users is not random means that if we do not choose proper frequency to pairs, then this will create interference between pairs and which means that communication between pairs is meaningless. Following figure 5 shows that giving same frequency f to both pairs create interference. Dotted line shows interference.

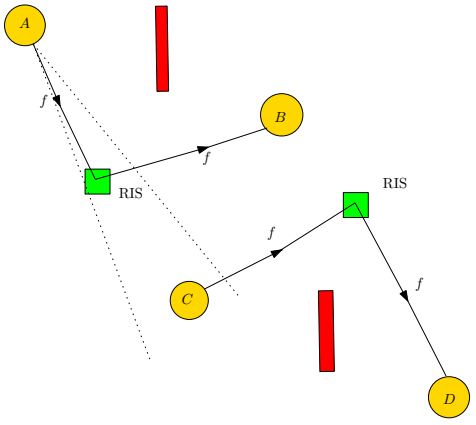


Figure 5

So when we choose RIS for one of the pair and allocate the frequency for that pair then after choosing RISs and frequencies for all pairs, there should not be

interference between pairs. So we have to choose RISs and allocate frequencies for the pairs such that after choosing this there will not be interference between pairs when they communicating.

For the communications of users, appropriate RIS should be selected as well as allocate frequency properly so that users communicate each other without interfering with other communicating pairs of users. But the selection of RIS and allocation of frequency should be considered simultaneously. Otherwise, there is a problem of interference. Consider the following example. In the following example, RISs R_1, R_2, R_3, frequency f are given. Also given that two communicating pairs, one pair is A, B and other pair is C, D which is shown in figure 1. In figure 2, without considering the frequency, RIS R_2 and R_3 are selected.Then after allocating frequency, there is interference between the pair C, D as shown in figure 3. So at the time of RISs selection, if frequency f teaking into consideration then RISs R_1 and R_3 are selected and allocated frequency f, then there is no interference between both the pairs and communication is possible as shown in figure 4.

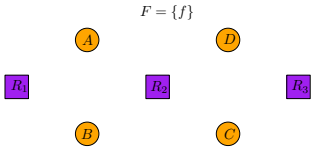


Figure 1

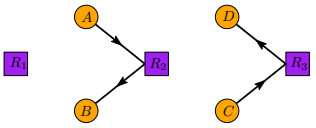


Figure 2

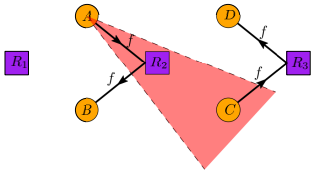


Figure 3

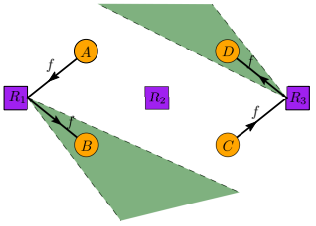


Figure 4

So RIS selection and frequency allocation are dependent on each other. Both has to be taken into consideration simultaneously which is hard to deal. This is matching problem and matching problem is NP-Hard problem [6]. So we will not get optimal solution for this pair but we can find approximate solution for this problem.

In this problem, for given set of pairs we have to serve maximum communicating pairs in available frequencies using RISs without interference.

Chapter 2

Related Works

Channel allocation problem for D2D communications is a well studied problem. In [7], the authors first select such D2D pairs which satisfying the QoS requirements and after that they allocate powers. Using maximum weight in bipartite matching, authors then allocate frequency to each selective D2D pair so that the overall system throughput should be maximized. In [8] this paper, authors have considered a resource allocation problem and considers maximizing throughput of D2D communication and cellular network subject to their minimum rate requirements. For the D2D links or pairs which are operating in the direct mode or relay node, optimal power allocations are calculated. Then, using these power allocations, the relay assignment problem has been solved as a job scheduling problem. Some of the few challenges in relay selection have been discussed in [1]. In [9] this paper, authors deal with a combined relay selection and resource allocation framework and gave greedy algorithm for solving this joint problem. In this algorithm, using SINR requirements authors first consider the assignment of frequency channels to the relays and then determines the optimal choice of relays. Therefore they deal the two problems in a two step manner. In [10] this paper, authors formulated a mixed integer non-linear programming problem for the combined power allocation and channel assignment in a D2D communication network where relays assist the communications and subsequently present two heuristic algorithms. In [11] this paper authors proposed a two step process for relay selection and resource allocation in which first using position of the candidate relays based on the sectored cell and after that they select relays and the corresponding frequencies.

In [12] this paper, relay assisted mmWave 5G cellular network is considered. In [13], first an non-linear integer programming is developed for the resource allocation problem, subsequently converted into binary integer programming using D2D clusters. Finally they solved the problem using branch and cut algorithm. In [14], the authors consider maximizing the system throughput in a relay-aided D2D communications. They solve the problem in four stages. Authors of [6] uses a linear program relaxation method to combinedly solve the relay selection and channel assignment problem. While in [5], the authors uses graph coloring and grouping based method to solve the same. Authors of [15] use similar approach and employ maximum independent set construction to deal the same.

The existing works, including the above ones either do not consider RIS aided communications or does not deal the problem of relay (RIS) selection and channel allocation jointly.

Chapter 3

System Model

In D2D communications, suppose there are n number of requesting or communicating pairs want to communicate. In this n requesting pairs, some requesting pairs are direct communicating pairs and some are communicating pairs which communicate via RISs. Suppose there are m number of RISs have been already placed around requesting pairs. Suppose there are k number of different frequencies.

Mobility Consideration: In this problem, we suppose that all the communicating pairs are stable. They are not in motion.

Reconfigurable Intelligent Surfaces(RIS): For simplicity assume that RIS can serve only one communicating pair. Suppose RISs have been already placed and there are sufficient RISs are placed or available around any communicating pairs. Suppose N and R be the set of communicating pairs and set of RISs respectively.

Frequency Allocation: Suppose F be the set of frequencies. We have to allocate available frequencies to the communicating pairs so that maximum number of pairs are served. One frequency can be give to the one or more communicating pairs.

Interference Consideration: If one frequency is given to two communicating pairs which is given in figure 6, the sender P is fall in that beam of frequency made by sender S and also sender P transmitting same frequency, in that scenario interference is considered.

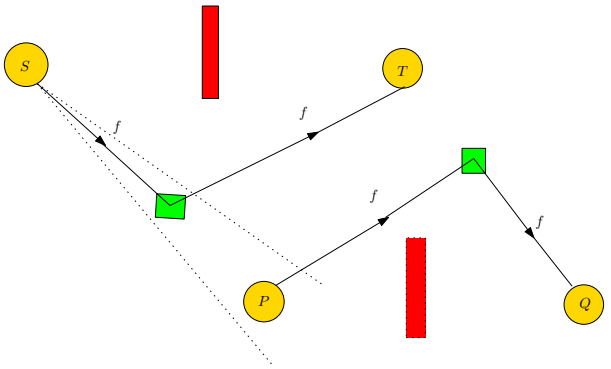


Figure 6

If two communicating pairs having two different frequencies fall in the beam of their respective frequencies, then this should not be considered as interference which is shown in figure 7.

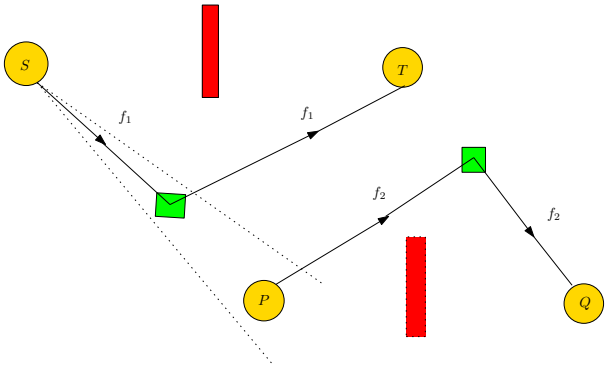


Figure 7

And if sender or receiver of pair having frequency f is not fall in beam of any other frequencies of pair then there is no interference in that pair.

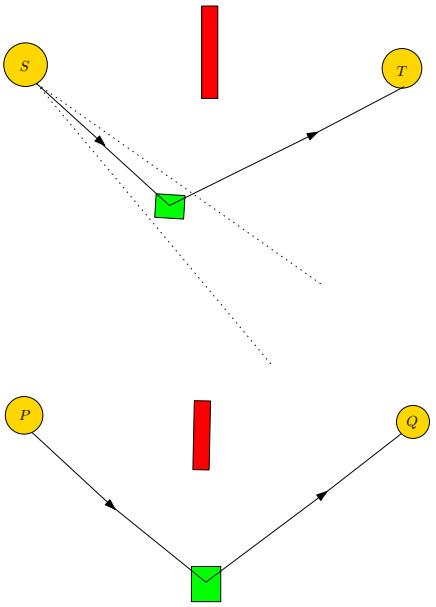


Figure 8

Chapter 4

Problem Formulation

Suppose X_{irf}, Y^d_{if}, Z_{ijf} are the indicator variables such that

$X_{irf} = 1$, if i^{th} communicating pair communicate using relay r using frequency f

$= 0$, Otherwise

$Y^d_{if} = 1$, if i^{th} communicating pair communicate directly using frequency f

$= 0$, Otherwise

and

Suppose C and C' are the sets such that C contains all communicating pairs which communicate using RISs and C' contains all communicating pairs which communicate directly and D be the union of these two sets. Let I be the set of interfering communicating pairs.
Let I_1 be the set of interfering communicating pairs such that for $(i,j) \in I_1$, both i^{th} and j^{th} pairs are direct communicating pairs. For this interfering pairs, constraint become,

$$Y^d_{if} + Y^d_{jf} \leqslant 1$$

Let I_2 be the set of interfering communicating pairs such that for $(i,j) \in I_2$, i^{th} pair communicate using RIS and j^{th} pair is direct communicating pair. For this interfering pairs, constraint become,

$$Y^d{}_{if} + \sum_{r \in R} X_{jrf} \leqslant 1$$

Let I_3 be the set of interfering communicating pairs such that for $(i, j) \in I_3$, both i^{th} and j^{th} pairs are communicating using RISs. For this interfering pairs, constraint become,

$$X_{irf} + \sum_{r \in R} X_{jrf} \leqslant 1$$

I_1, I_2 and I_3 are subsets of I.
Therefore the problem is,

$$\textbf{max} \sum_{i \in C} \sum_{r \in R} \sum_{f \in F} X_{irf} + \sum_{i \in C'} \sum_{f \in F} Y^d{}_{if}$$

subject to

$$\sum_{r \in R} \sum_{f \in F} X_{irf} \leqslant 1, \forall i \in C$$

$$\sum_{f \in F} Y^d{}_{if} \leqslant 1, \forall i \in C'$$

$$Y^d{}_{if} + Y^d{}_{jf} \leqslant 1, \forall (i, j) \in I_1$$

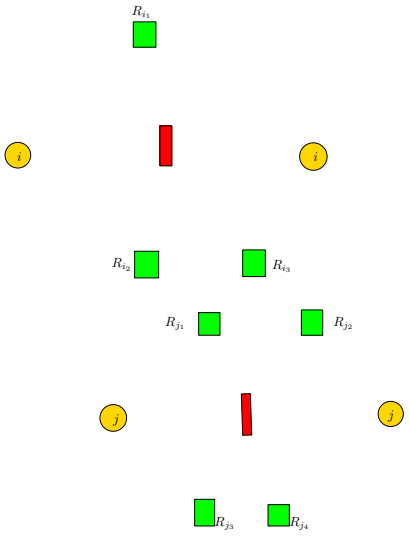
$$Y^d{}_{if} + \sum_{r \in R} X_{jrf} \leqslant 1, \forall (i, j) \in I_2$$

$$X_{irf} + \sum_{r \in R} X_{jrf} \leqslant 1, \forall (i, j) \in I_3$$

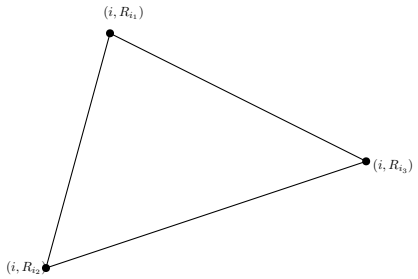
Chapter 5

Problem Solution

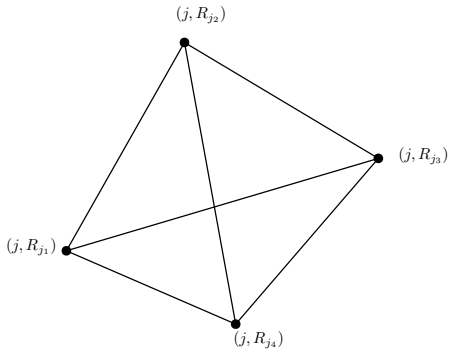
Suppose one requesting pair i in D want to communicate.Here requesting pair i means it contains both the users which want to communicate each other.Then consider all the RISs around that pair i which can serve pair i. Suppose in particular there are 3 RISs around pair i which can serve pair i and they are R_{i_1}, R_{i_2} and R_{i_3}. Similarly, for the pair j in D which can serve pair j. Suppose in particular there are 4 RISs around pair j which can serve pair j and they are $R_{j_1}, R_{j_2}, R_{j_3}$ and R_{j_4} which is shown in following figure,



Consider (i, R_{i_1}), (i, R_{i_2}), (i, R_{i_3}) as a nodes and make a clique as following



Consider (j, R_{j_1}), (j, R_{j_2}), (j, R_{j_3}), (j, R_{j_4}) as a nodes and make a clique as following



Consider the following Figure 9(a) and Figure 9(b), in these figures both the users of pair i denoted by i and also both the users of pair j denoted by j. Now consider the Figure 9(a),in this figure pair i communicate via RIS R_{i_2} create interference with pair j which communicate via RIS R_{j_1}. Also in the Figure 9(b),in this figure pair i communicate via RIS R_{i_3} create interference with pair j which communicate via RIS R_{j_2}. And in the Figure 9(c),in this figure pair j communicate via RIS R_{j_1} create interference with pair i which communicate via RIS R_{i_3}.

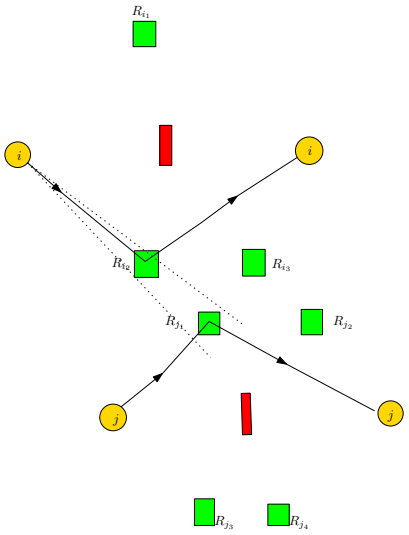


Figure 9 (a)

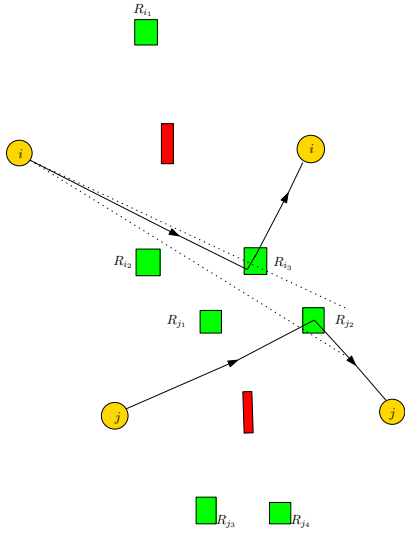


Figure 9(b)

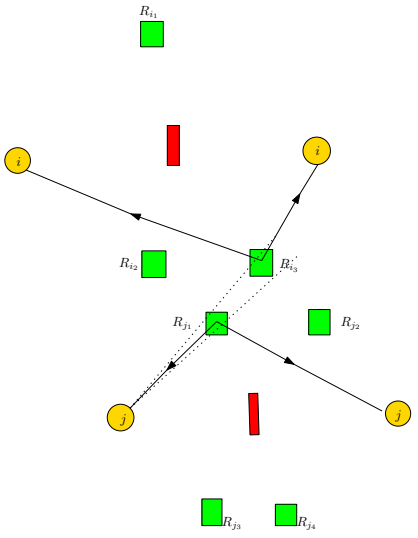
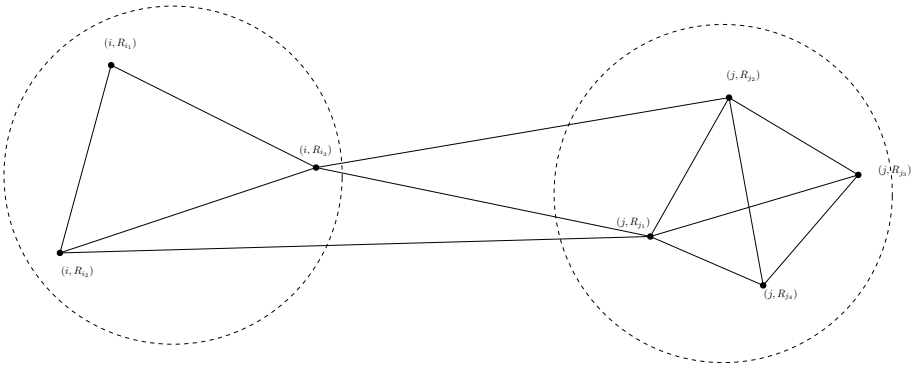


Figure 9(c)

Using above information we make interference graph. In this graph vertices are cliques. There is an edge between nodes if and only if pair i which communicate via RIS interfere with other communicating pair j.For example in Figure 9(a), pair i communicate via RIS R_{i_2} create interference with pair j which communicate via RIS R_{j_1}. So there is an edge between node (i, R_{i_2}) and (j, R_{j_1}). Also in the Figure 9(b), pair i communicate via RIS R_{i_3} create interference with pair j which communicate via RIS R_{j_2}. So there is an edge between node (i, R_{i_3}) and (j, R_{j_2}). And similarly from Figure 9(c), there is an edge between node (i, R_{i_3}) and (j, R_{j_1}).So interference graph formed is as follows,



Therefore our problem becomes interference graph with vertices as a clique and we want to select one node from one clique and assign colour(frequency) for that node which implies one RIS and frequency selected for one communicating pair.

Since we have to select only one RIS from one clique, we want to give only one frequency to one clique. So we convert this problem into as graph coloring problem. For that treat frequency as a colour and suppose that edge weight for the edges in clique be infinite and edge weight for the edges out of clique be 1. We consider maximum weights for clique because if one colour gave to one node in clique then no colour is assigned to other node. This means that only one colour is assigned to one clique which further implies for one communicating pair one RIS is selected and one frequency is assigned.

So basically this is a graph colouring problem and there are many coloring algorithm. For coloring algorithm, our input graph is interference graph which contain cliques as a vertices

$V_1 = \{(1, R_{11}), (1, R_{12}), ..., (1, R_{1k_1})\}$

$V_2 = \{(2, R_{21}), (2, R_{22}), ..., (2, R_{2k_2})\}$

$\vdots$

$V_l = \{(l, R_{l1}), (l, R_{l2}), ..., (l, R_{lk_l})\}$

These are the l communicating pairs which communicates using RIS. For the communicating pairs which communicates directly, for these pairs consider dummy RISs. So for any $j \in C'$ pair, clique contain only one node of the form $V_j = (j, R_j)$

There are n communicating pairs so there are n cliques.

Chapter 6

Algorithm

For graph coloring we will consider the parameters minimum clique count, minimum degree of vertex, minimum cardinality of vertex and maximum wtsum which is weight sum of vertex. We consider minimum clique count because minimum clique count implies members in the clique is minimum which further implies that number of RISs for that pair is minimum. And we want to serve all the pairs or maximum pairs so first consider the pairs which have minimum number of RISs. Minimum degree also implies same.But minimum clique count and minimum degree does not give redundancy. So we consider both these parameters. We consider minimum cardinality of vertex because cardinality represents distinct edge weights if cardinality is minimum which implies vertex has either edges with maximum weight or minimum weight 1. If vertex has only maximum weight which means that this vertex does not interfere with other vertex and we can color this vertex and if vertex has only mimimum weights 1, this means this vertex is a pair with only one RIS and interfering with other vertices so this vertex should be taken into consideration. We consider maximum weight sum because maximum weight sum implies that this vertex interfere with minimum number of other vertex and we can color this vertex.

1. **Algorithm 1:Assign RIS frequency**
2. **Algorithm 2:getDegCardWtsumCq**
3. **Algorithm 3:evaluate_min**

4. **Algorithm 4:evaluate_max**
5. **Algorithm 5:Modify**
6. **Algorithm 6:colorcheck**
7. **Algorithm 7:update adjG basis cInfo**

Algorithm 1 Assign RIS frequency

Require: $G = (V, E)$; $V_{ij} = (C_i, R_j)$; C_i = comm pair, R_j = RIS available

Ensure: RIS frequency assignment

Step 1:
for all $(V_{ij}, V_{pq}) \in V \times V$ **do**
  if $V_{ij} \in \text{Clique}(V_{pq})$ **then**
    $E(V_{ij}, V_{pq}) \leftarrow E_{max}$
  else if V_{ij} is interfering with V_{pq} **then**
    $E(V_{ij}, V_{pq}) \leftarrow 1$
  else
    $E(V_{ij}, V_{pq}) \leftarrow 0$
  end if
end for
Step 2:
Initialize:
$V'[1 : n] \leftarrow$ sorted array of V_{ij}
assignment$[i] \leftarrow -1 \quad \forall i$ from 1 to $n = |V'|$
adjG$[1 : n, 1 : n] \leftarrow$ adjacency matrix of G
basis$[i, j] \leftarrow 1$ if adjG$[i, j] \neq 0 \quad \forall i, j$; 0 otherwise
cInfo$[i, j] \leftarrow 1$ if $V'_i \in \text{Clique}(V'_j) \quad \forall i, j$; 0 otherwise
clique list$[1 : m] \leftarrow$ array of distinct C_i
nCliques $\leftarrow$ len(clique list)
clique mem$[i] \leftarrow$ array of $(C_i, R_j) \quad \forall i \in$ clique list
clique count$[i] \leftarrow$ count(clique mem[j]) $\quad \forall i$ and $V'_i = V_{jk}$
deg$[i] \leftarrow$ out-degree of $V'_i \quad \forall i$
card$[i] \leftarrow$ count(distinct $E(V'_i, V'_k)$) $\quad \forall k$
wtSum$[i] \leftarrow$ sum$(E(V'_i, V'_k)) \quad \forall k \in$ In
colors $\leftarrow$ array of available frequencies
exhausted $\leftarrow []$
Step 3:

```
28: for i = 1 to nCliques do
29:   if colors is non-empty then
30:     deg, card, wtSum, clique count ← getDegCardWtsumCq(adjG, basis, cInfo, n)

31:     (x, vIndex) ← evaluate min(clique count, V′)
32:     if x = 1 then
33:       exhausted ← modify(vIndex, exhausted, G, clique mem, colors)
34:     else if x > 1 then
35:       min set ← array of all nodes with same clique count as
          clique count[vIndex]
36:       (x, vIndex) ← evaluate min(deg, min set)
37:       if x = 1 then
38:         exhausted ← modify(vIndex, exhausted, G, clique mem, colors)
39:       else if x > 1 then
40:         min set ← array of all nodes with same degree as deg[vIndex]
41:         (x, vIndex) ← evaluate min(card, min set)
42:         if x = 1 then
43:           exhausted ← modify(vIndex, exhausted, G, clique mem, colors)
44:         else if x > 1 then
45:           min set ← array of all nodes with same cardinality
46:           (x, vIndex) ← evaluate max(wtSum, min set)
47:           exhausted ← modify(vIndex, exhausted, G, clique mem, colors)
48:         end if
49:       end if
50:     else
51:       Output: Insufficient colors available, insufficient number of RISs
52:       Exit
53:     end if
54:   end if
55: end for
56: for i = 1 to n do
57:   if assignment[i] ≠ −1 then
58:     Output: (V_pq = V′[i], assignment[i])
59:   end if
60: end for
```

Explanation of algorithm: In this algorithm we color interference graph.

For this first consider Algorithm assign_RIS_freq. In this we initialize graph as $G =< V, E >$; $V_{ij} = (C_i, R_j)$; Ci = communicating pair, R_j = RIS which are available for that pair. Here V and E are vertices and edges of graph.

In (1), algorithm states that if vertices belong to same clique then give edge weights E_max and if vertices are not belongs to clique means V_{ij} is interfering with V_{pq} then give edge weights 1 and otherwise give weights 0. In (2),algorithm initializes some arrays.$V'[1:n]$ is sorted array of V_{ij}. First initialize all the values in sorted by -1. Then consider the adjacency matrix adjG$[1:n, 1:n]$ of G which contains edge weights.

basis$[i, j]$ is a matrix in which entry is 1 if adjG$[i, j] \neq 0 \forall i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, n;\ 0$, otherwise which means that entry of basis$[i, j]$ indicates that if there is an edge between vertices then give entry 1.

cInfo$[i, j]$ is a matrix which contains information of cliques. In this matrix entry is 1 if $V'_i \in Clique(V'_j)\ \forall i = 1, 2, \ldots, n;\ j = 1, 2, ..., n;\ 0$, otherwise which means that if vertex belong to clique of other vertex implies that this vertex is also member of that clique.

clique_list$[1:m]$ is an array of distinct C_i

e.g. i, j, k

nCliques is length of clique_list

clique_mem[i] is an array of $(C_i, R_j)\ \forall i \in$clique_list

e.g. clique_mem[v] = $[(i, 1), (i, 2)]$ 1, 2 means name of RISs

clique_count$[i]$ is an array whose entry contains number of members in that clique and which is calculated by count(clique_mem[j]) $\forall i = 1, 2, \ldots, n$ and $V'[i] = V_{jk}$

deg$[i]$ is an array of out-degree of vertices in $V'_i\ \forall i = 1, 2, \ldots, n$

card$[i]$ denotes the number of distinct edge weights for vertex and is calculated by count(distinct $E(V'_i, V'_k))\ \forall k$

wtSum$[i]$ denotes the sum of edge weights for vertex and is calculated by sum$(E(V'_i, V'_k))\ \forall k \in I_n$

colors[ ] is an array of available frequencies

exhausted [ ] is an array used for iterations.

These are the all initialization for graph coloring.

in step (3) of Algorithm assign_RIS_freq for $i = 1, 2, 3, \ldots, n$ Cliques ,if colors array is non-empty deg, card, wtSum, clique_count is

getDegCardWtsumCq(adjG, basis, cInfo, n) gives value of degree= deg, card=cardinality, wtsum= weightsum, clique_count of vertices.

$(x, vIndex)$ is evaluate_min(clique_count, V') indicates that x contains number of

times minimum value is repeated among clique count and vIndex gives index of that vertex which have minimum clique count
if x is 1 which means that clique count is unique and so we get vertex for coloring then we do not consider this vertex for next iteration and such vertex store in exausted. so we get exhausted = modify(vIndex, exhausted, G, clique_mem, colors)
else if $x > 1$ means that clique count is not unique then go for the parameter degree. For that consider the set min_set which is array of all nodes with same clique_count[vIndex]. We now consider degree parameter for min_set
Now $(x, vIndex)$ is evaluate_min(deg, min_set) means x contains number of times minimum value is repeated among degree.
if x is 1 means that degree is unique and so we get vertex for coloring then we do not consider this vertex for next iteration and such vertex store in exausted. so we get exhausted = modify(vIndex, exhausted, G, clique_mem, colors)
else if $x > 1$ which means degree is not unique then consider min_set which is an array of all nodes with samedegree[vIndex]. Now $(x, vIndex)$ is evaluate_min(card, min_set) means x contains number of times minimum value is repeated among cardinality. If x is 1 means that cardinality is unique and so we get vertex for coloring then we do not consider this vertex for next iteration and such vertex store in exausted. so we get exhausted = modify(vIndex, exhausted, G, clique_mem, colors)
else if $x > 1$ which means cardinality is not unique then consider min_set is an array of all nodes with same cardinality. Now $(x, vIndex)$ is
evaluate_max(wtSum, min_set) means x contains number of times maximum value is repeated among wtsum. If x is 1 means that wtsum is unique and so we get vertex for coloring then we do not consider this vertex for next iteration and such vertex store in exausted. so we get exhausted = modify(vIndex, exhausted, G, clique_mem, colors)
else Output is Insufficient colors available or insufficient number of RISs and algorithm exits.
Time complexity: We will compute time complexity of this algorithm after computing time complexity of all other algorithm.

Algorithm 2 getDegCardWtsumCq

Require: adjG, basis, cInfo, n
Ensure: deg, card, wtSum, cliquecount
1: **for** $j = 1, 2, \ldots, n$ **do**
2: temp $\leftarrow \emptyset$
3: deg[j] $\leftarrow 0$
4: card[j] $\leftarrow 0$
5: wtSum[j] $\leftarrow 0$
6: clique count[j] $\leftarrow 0$
7: **for** $i = 1, 2, \ldots, n$ **do**
8: **if** (basis[i, j] is not zero) **then**
9: deg[j] $\leftarrow$ deg[j] $+ 1$
10: temp $\leftarrow$ temp $\cup$ adjG[i, j]
11: wtSum[j] $\leftarrow$ wtSum[j] $+$ adjG[i, j]
12: **if** (cInfo[i, j] is not zero) **then**
13: clique count[j] $\leftarrow$ clique count[j] $+ 1$
14: **end if**
15: **end if**
16: **end for**
17: card[j] $\leftarrow$ |temp|
18: **end for**
19: **return** deg, card, wtSum, clique count

Explanation of algorithm: The algorithm takes the input matrices adjG, basis, and cInfo, along with the value n. It initializes variables temp, deg[j], card[j], wtSum[j], and cliquecount[j] to zero for each column j in the range 1 to n. It iterates over each column j from 1 to n. Inside the outer loop, a temporary set temp is created to store unique values. The variables deg[j], card[j], wtSum[j], and cliquecount[j] are set to zero for the current column j. It then enters an inner loop that iterates over each row i from 1 to n. Inside the inner loop, it checks if the element basis[i, j] is non-zero. If basis[i, j] is non-zero, it increments deg[j] by 1, adds the corresponding element adjG[i, j] to temp, and updates wtSum[j] by adding adjG[i, j]. Additionally, if the element cInfo[i, j] is non-zero, it increments cliquecount[j] by 1. After the inner loop completes, the algorithm calculates the cardinality of temp (the number of unique elements in temp) and assigns it to card[j]. The outer loop continues to the next column j and repeats steps 4 to 10 until all columns are processed. Finally, the algorithm returns the calculated values deg, card, wtSum, and cliquecount.

Algorithm 3 evaluate_Min

Require: $array1$, $array2$
Ensure: $nmin$, j
1: min val, $j \leftarrow (\min(array1[i]), i)$ for each i in indices($array2$)
2: $nmin \leftarrow 0$
3: **for** each i in indices(v') **do**
4: **if** $array1[i]$ is min val **then**
5: $nmin \leftarrow nmin + 1$
6: **end if**
7: **end for**
8: **return** $nmin$, j

Explanation of algorithm: The algorithm evaluate_Min aims to find the minimum value in array1 and its corresponding index in array2. It then counts the number of elements in array1 that have the minimum value. Here is an explanation of the algorithm step by step: In line 1, the algorithm initializes the variables min val and j by finding the minimum value in array1 using the min function. The min function returns the minimum value and its corresponding index in array2. In line 2, the variable nmin is initialized to zero. This variable will be used to count the number of elements in array1 that have the minimum value. The algorithm enters a loop starting from line 3, iterating over each index i in array2. In line 4, the algorithm checks if the element array1[i] is equal to the min val, which represents the minimum value in array1. If it is, this means that array1[i] has the minimum value. If the condition in line 4 is true, the algorithm increments the nmin variable by 1 in line 5, indicating that another element in array1 has the minimum value. The loop continues to the next index i until all elements in array2 have been processed. After the loop finishes, the algorithm returns the values of nmin and j in line 8 as the output of the algorithm. In summary, the algorithm finds the minimum value in array1 and its corresponding index in array2. It then counts the number of elements in array1 that have the minimum value. The output of the algorithm is the count nmin and the index j associated with the minimum value. The algorithm assumes that array1 and array2 are input arrays of the same length.

Time complexity: To find evaluate_min algorithm takes O(n) time because since the size of array2 can atmost be n, finding the minimum value in this step takes O(n) time. Further, finding the number of such minimum values obtained can also be done simultaneously or requires another traversal through the array. Altogether, this evaluate_min step takes O(n) time.

Algorithm 4 evaluate_Max

Require: $array1$, $array2$
Ensure: $nmax$, j
1: max val, $j \leftarrow (\max(array1[i]), i)$ for each i in indices($array2$)
2: $nmax \leftarrow 0$
3: **for** each i in indices($array2$) **do**
4: **if** $array1[i]$ is max val **then**
5: $nmax \leftarrow nmax + 1$
6: **end if**
7: **end for**
8: **return** $nmax$, j

Explanation of algorithm: The evaluateMax algorithm takes two input arrays, array1 and array2, and aims to find the maximum value in array1, identify its corresponding index in array2, and count the number of occurrences of the maximum value in array1. The algorithm follows the steps outlined below: It initializes the variables max_val and j to keep track of the maximum value and its corresponding index, respectively. The assignment (max(array1[i]), i) finds the maximum value in array1 and returns its index in array2, which is stored in j. The variable nmax is initialized to zero to count the number of occurrences of the maximum value in array1. The algorithm enters a loop that iterates over each index i in array2. Within the loop, the algorithm checks if array1[i] is equal to the maximum value max_val. If this condition is true, it means that array1[i] is one occurrence of the maximum value. If array1[i] is indeed the maximum value, the algorithm increments nmax by 1. After checking each element in array2, the loop terminates. Finally, the algorithm returns the values of nmax and j as the output. The purpose of the algorithm is to identify the maximum value in array1, find its corresponding index in array2, and count the number of occurrences of the maximum value in array1. Note that the algorithm assumes that max(array1[i]) returns the maximum value in array1 and (max(array1[i]), i) returns the index in array2 corresponding to the maximum value. In conclusion, the evaluateMax algorithm correctly identifies the maximum value in array1, determines its corresponding index in array2, and counts the number of occurrences of the maximum value in array1.

Time complexity: Algorithm evaluate_max(array1, array2) takes O(n) time because since the size of array2 can atmost be n, finding the maximum value in

this step takes O(n) time. Further, finding the number of such maximum values obtained can also be done simultaneously or requires another traversal through the array. Altogether, this evaluate_max step takes O(n) time.

Algorithm 5 Modify

Require: $vIndex$, G, $cliquemem$, $colors$
Ensure: $exhausted$
1: node $\leftarrow$ $Vpq = V'$[vIndex]
2: new exhausted $\leftarrow$ node $\cup$ {array of nodes in V forming clique with the node}
3: or having RIS Rq
4: assignment[vIndex] $\leftarrow$ colors[0]
5: exhausted $\leftarrow$ exhausted $\cup$ new exhausted
6: colors, basis $\leftarrow$ colorcheck(exhausted, vIndex, basis, G, n, colors)
7: update adjG basis cInfo(new exhausted, adjG, basis, cInfo, n)
8: **return** exhausted

Explanation of algorithm: The Modify algorithm takes several inputs, including vIndex, G, clique_mem, and colors, and performs modifications to update the state of the algorithm. The algorithm follows the steps outlined below: It assigns the value of Vpq = V' [vIndex] to the variable node. This represents a specific node in the graph. The algorithm generates a new set of nodes called new_exhausted. This set includes the nodes in V that form a clique with the node or (RIS) Rq. In other words, it includes the nodes that are adjacent to node or have a relationship with node according to the problem's criteria. The algorithm assigns the color colors[0] to the variable assignment[vIndex]. This indicates the color assigned to the specific node. The exhausted set is updated by adding the nodes from the new_exhausted set. This ensures that the newly added nodes are marked as exhausted. The algorithm calls the colorcheck function, passing the updated exhausted set, vIndex, basis, G, n, and colors as parameters. This function performs checks and modifications related to coloring and updates the colors and basis accordingly. The adjG, basis, and cInfo matrices are updated using the update function, which takes the new_exhausted set, adjG, basis, cInfo, and n as parameters. This update ensures that the relationships and information related to the newly added nodes are reflected in these matrices. Finally, the algorithm returns the updated exhausted set as the output. The purpose of the Modify algorithm is to modify and update various data structures and variables based on the specific node 'node' and its relationships in the graph. This ensures that the algorithm progresses correctly and maintains consistency in its operations. Note that the exact details of

the 'colorcheck' and 'update' functions are not provided in the algorithm, but they are assumed to perform the necessary checks and updates as mentioned in their respective descriptions. In conclusion, the Modify algorithm correctly modifies and updates the state of the algorithm by assigning colors, updating the exhausted set, performing color checks, and updating relevant matrices based on the specific node and its relationships.

Time complexity: In Evaluating new_exhausted requires us to traverse through all the vertices in the given graph. This takes O(n) time. Further, colorcheck takes $O(n^2)$ time. Peforming update_adjG_basis_cInfo can be done in O(n) time. Thus, this operation takes O(n2) time and higher complexity in this step is due to colorcheck operation.

Algorithm 6 colorcheck

Require: exhausted, vIndex, basis, G, n, colors
Ensure: colors, b
temp count $\leftarrow 0$
for $i = 1, 2, \ldots, n$ **do**
  if V'[i] is not in exhausted and basis[vIndex][i] $= 0$ **then**
    return colors, b
  else
    temp count $\leftarrow$ temp count $+ 1$
  end if
end for
if temp count $+$ len(exhausted) == n **then**
  if colors is not empty **then**
    delete colors[0]
  else
    Output: Insufficient colors available **or** Insufficient No. of RIS
    Exit
  end if
end if
return colors, b

Explanation of algorithm: The colorcheck algorithm is responsible for performing checks and modifications related to coloring. It takes several inputs, including exhausted, vIndex, basis, G, n, and colors, and updates the colors array accordingly. The algorithm follows the steps outlined below: The variable temp_count is initialized to zero. This variable will keep track of the number of nodes that satisfy

certain conditions. The algorithm iterates over each node V'[i] in the range of 1 to n. If the node V'[i] is not in the exhausted set and the value of basis[vIndex][i] is 0, it means that the node V'[i] is a valid candidate for coloring. In this case, the algorithm immediately returns the colors array and b. If the condition in step 3 is not met, it means that the node V'[i] is not a valid candidate for coloring. In this case, the temp_count variable is incremented. After iterating over all nodes, the algorithm checks if the sum of temp_count and the length of the exhausted set is equal to n. This condition ensures that all nodes have been considered for coloring. If the condition in step 5 is true, it means that there are no remaining nodes available for coloring. In this case, the algorithm checks if the colors array is empty. If the colors array is not empty, it means that there are still available colors. In this case, the algorithm deletes the first element of the colors array. If the colors array is empty, it means that there are no more available colors. The algorithm outputs a message indicating that there are insufficient colors available or insufficient number RIS and exits. Finally, the algorithm returns the updated colors array and b. The purpose of the colorcheck algorithm is to check if there are any valid candidates for coloring among the remaining nodes for already used color. Means that color should be retained or not. It ensures that all nodes are considered and, if necessary, adjusts the colors array by removing a color if it becomes unavailable. Additionally, it handles the case where there are no more available colors or RISs. Note that the exact purpose and meaning of the b variable are not explicitly mentioned in the algorithm. It is assumed to have a specific purpose within the broader context of the algorithm. In conclusion, the colorcheck algorithm correctly performs checks related to coloring by examining the exhausted set, basis matrix, and colors array. It ensures that valid candidates for coloring are identified and adjusts the colors array accordingly, while also handling the case of insufficient available colors or RISs.

Time complexity: Performing colorcheck requires us to loop through all the vertices in the given graph. Further, each such vertex, we have to examine the exhausted array whose size is atmost n. Thus, performing colorcheck takes atmost nxn i.e., $O(n^2)$.

Algorithm 7 update adjG basis cInfo

Require: $exhausted$, $adjG$, $basis$, $cInfo$, n
Ensure: None
1: **for** each $V'_p = V_{ij}$ in $exhausted$ **do**
2: **for** $k = 1, 2, \ldots, n$ **do**
3: $adjG[i][k] = 0$
4: **end for**
5: **for** $k = 1, 2, \ldots, n$ **do**
6: $adjG[k][j] = 0$
7: **end for**
8: $basis[i][j] = n + 1$
9: $cInfo[i][j] = 0$
10: **end for**

Explanation of algorithm: The update adjG basis cInfo algorithm is responsible for updating the adjG, basis, and cInfo matrices based on the nodes in the exhausted set. The algorithm follows the steps outlined below: The algorithm iterates over each node V'[p] = V[i][j] in the exhausted set. For each node V'[p], the algorithm iterates over the range of k from 1 to n. Within the inner loop, the algorithm sets the value of adjG[i][k] to 0. This operation effectively removes any existing edges between the node V'[p] and other nodes in the adjG matrix. After updating the adjG matrix for the current V'[p], the algorithm proceeds to the next loop. Similarly, the algorithm iterates over the range of k from 1 to n. Within this second inner loop, the algorithm sets the value of adjG[k][j] to 0. This operation removes any existing edges between other nodes and the node V'[p]. The algorithm then updates the basis matrix by setting basis[i][j] to n + 1. This update effectively marks the edge between the nodes V'[p] and V[i][j] as non-existent by assigning a value greater than n to it. Finally, the algorithm updates the cInfo matrix by setting cInfo[i][j] to 0. This update signifies that there is no longer any connection or clique relationship between the nodes V'[p] and V[i][j]. The algorithm repeats steps 2 to 8 for each node V'[p] in the exhausted set. The purpose of the update adjG basis cInfo algorithm is to remove edges and update matrices based on the nodes in the exhausted set. By setting the corresponding elements in the adjG, basis, and cInfo matrices to appropriate values, the algorithm effectively eliminates connections and relationships involving the nodes in the exhausted set. Note that the algorithm assumes that adjG, basis, and cInfo are matrices or data structures that can be accessed and modified using the given indices and assignment statements. In conclusion, the update adjG basis cInfo algorithm correctly

updates the adjG, basis, and cInfo matrices by removing edges and updating values based on the nodes in the exhausted set, as specified in the steps of the algorithm.

Time complexity Algorithm update_adjG_ basis_cInfo(exhausted, adjG, basis, cInfo, n) takes O(n) time because updating the adjG, basis and cInfo matrices require us to traverse through exhausted array. The length of exhausted array is less than n. Thus this procedure takes O(n) time.

Lastly, after combining all these algorithms, main algorithm return vertex with appropriate color.

Time complexity of Algorithm assign_RIS_freq
Let n represent the number of vertices in G step-1 takes $O(n^2)$ as it has to loop through all the pairs of vertices for assigning edge weight
Since the sorting of the vertices is arbitrary and any such order can be considered, initialization of the vertices takes O(n) time.
since the assignment of -1 is done to every vertex in the given graph, assignment takes O(n) time.
Let E represent no. Of edges. E is of the $O(n^2)$. So, the construction of adjacency matrix and basis can be done simultaneously and requires $O(n^2)$ time

Since an edge of Emax represents an edge between members belonging to same clique, construction of cInfo requires O(E) = $O(n^2)$ time. Construction of clique_list requires us to traverse through the array of vertices and examine the communication pair as Vij = (Ci, Rj). While traversing through the array, distinct communication pair Ci and their count, their corresponding members can be tracked. so, construction of clique_list, clique_mem and clique_count together takes O(n) time. Construction of deg, card and wtSum matrices or 2D arrays requires us to traverse throughthe adjG matrix whose size is n^2 and can be done together, thus construction of these datastructures take $O(n^2)$time. Fetching deg, card, wtSum and clique_count information requires $O(n^2)$ time. Performing evaluate_min takes O(n) time.
Performing modify operation requires $O(n^2)$ time.Remaining step of algorithm requires us to traverse through the assignment array whose size is the size of vertices array. Thus, this step takes O(n) time.

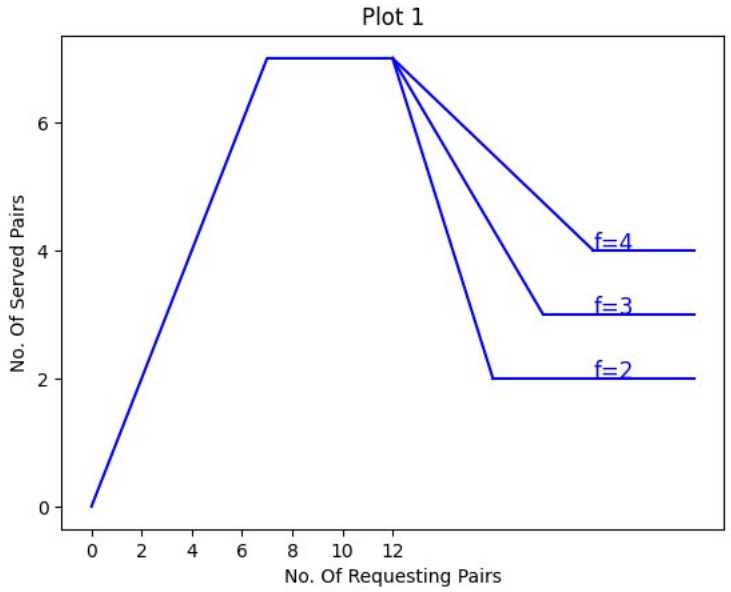
Therefore the above algorithm takes $O(|Ci|n^2)$ time. Since the number of cliques is usually far less than the number of vertices in the graph, |Ci| can be treated as a constant.Thus the complexity of the above algorithm is $O(n^2)$.

Chapter 7

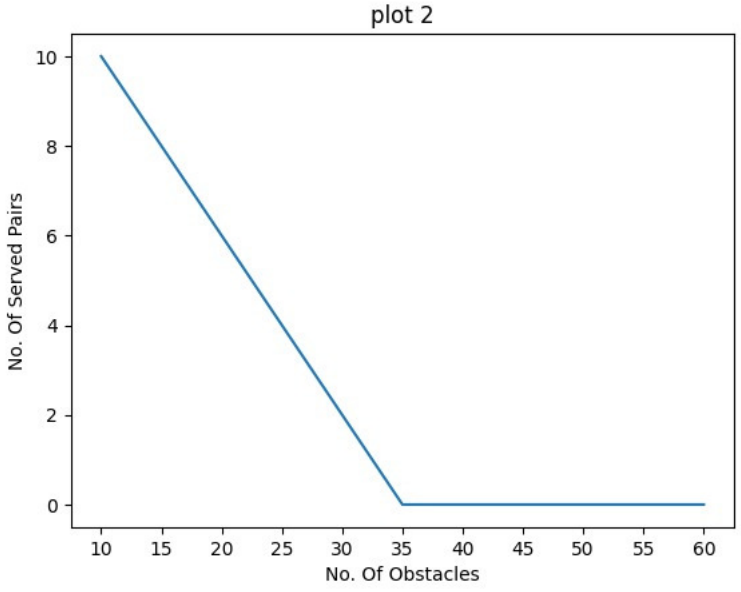
Experiment and Results

For the experiment, we consider a simulation environment similar to [5, 14] and compare our proposed method against a two-step method [14]. We consider the area or region of size 1000m×1000m. Direct communication is possible between the users if distance between the users is 50m. So consider the requesting pairs and available RISs in region of 1000m×1000m. And size of obstacle is fixed.

For the Plot 1, first suppose that number of RISs is fixed which is 10 and plot the graph No. of served pairs(Number of served pairs) vs No. of requesting pairs(Number of requesting pairs) using frequency $f = 2$, $f = 3$,$f = 4$. By considering number of examples,in which RISs are at fixed positions and obstacles are also at fixed position.Also suppose that requesting pairs are around some RISs not around all RISs then we get the same curve for $f = 2$, $f = 3$,$f = 4$. But if more requesting pairs are coming around these some RISs then interference links increases so for $f = 2$ graph behave first straight line $y = x$ through origin and then it behaves constant line and then after some point graph decreases and become line $y = 2$ because more requesting pairs means more interference and so $f = 2$, 2 frequencies can serve only two pairs so graph behave like line $y = 2$. Similarly for $f = 3$ graph behave first straight line $y = x$ through origin and then it behaves constant line and then after some point graph decreases and become line $y = 3$ because more requesting pairs means more interference and so $f = 3$, 3 frequencies can serve only three pairs so graph behave like line $y = 3$.
And also for $f = 4$ graph behave first straight line $y = x$ through origin and then it behaves constant line and then after some point graph decreases and become line $y = 4$ because more requesting pairs means more interference and so $f = 4$, 4 frequencies can serve only four pairs so graph behave like line $y = 4$.

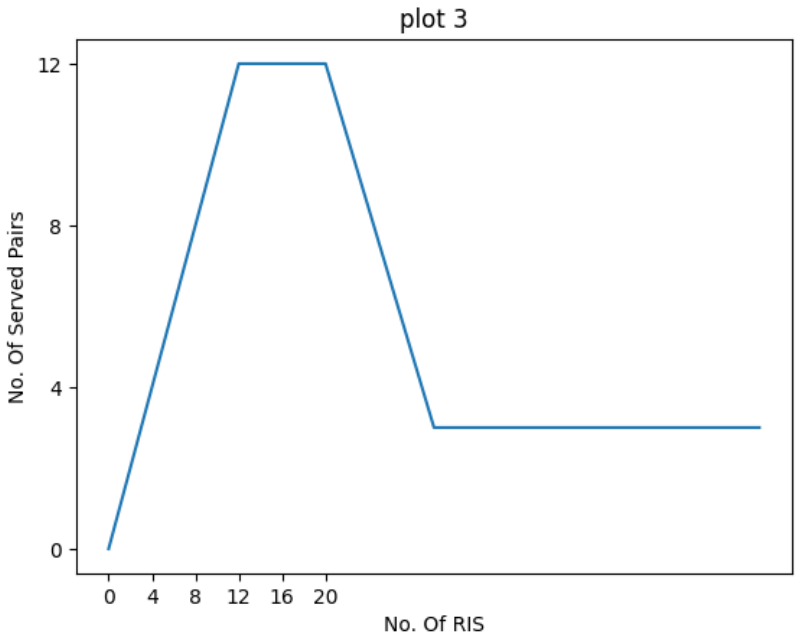


For plot 2, suppose that number of RISs is fixed which is 10 and placed at proper position and number of frequencies is 2. Then plot the graph No. of served pairs vs No. of obstacles. Plot 2 shows that as number of obstacles increases, number of served pair decreases.

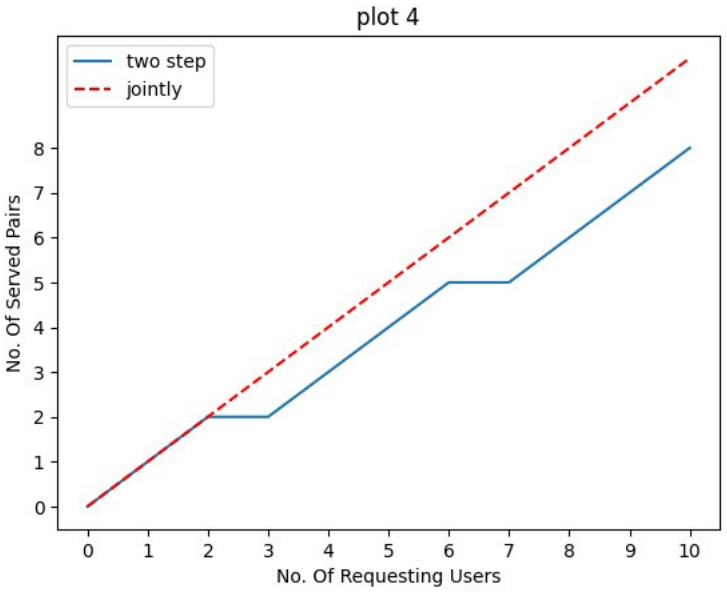


For plot 3, suppose that number of frequencies is fixed which is 3. Obstacles are at fixed position. Then plot the graph No. of served pairs vs No. of RIS. In this plot first graph behave like straight line $y = x$ then become parallel to x- axis and then decreases and become parallel to x-axis. This is because if considering situation

as requesting pairs are around only some RISs and as requesting pairs increases interference is also increases and since we have only 3 frequencies so only 3 pairs will be served and graph become straight line $y = 3$.



For plot 4, number of RISs is fixed which is 10. Obstacles are at fixed position and consider $f = 3$. Plot the graph No. of served pairs vs No. of requesting pairs. If we consider jointly RIS selection and frequency allocation then we get red curve and if we consider two step algorithm means first select RIS and then allocate frequency then we get blue curve. In two step, RIS which is nearest to requesting pair is selected. Plot shows that joint algorithm gives better result than two step algorithm.


plot 4
two step
jointly
No. Of Served Pairs
No. Of Requesting Users
0
1
2
3
4
5
6
7
8
9
10

Bibliography

[1] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, pp. 86–92, May 2014.

[2] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5g cellular: It will work!," *IEEE Access*, vol. 1, pp. 335–349, 2013.

[3] "Analysis of 28ghz and 60ghz channel measurements in an indoor environment (telecom infra project)."

[4] Ö. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling," *IEEE Wireless Communications Letters*, vol. 9, pp. 581–585, May 2020.

[5] R. N. Dutta and S. C. Ghosh, "Resource allocation for millimeter wave d2d communications in presence of static obstacles," in *Proceedings of the **35th International Conference on Advanced Information Networking and Applications (AINA-2021)**, Toronto, ON, Canada, 12-14 May, 2021*, vol. 225 of *Lecture Notes in Networks and Systems (LNNS)*, pp. 667–680, Springer, 2021.

[6] R. N. Dutta and S. C. Ghosh, "Joint relay selection and frequency allocation for D2D communications," in *Proceedings of the **17th EAI International Conference on Quality, Reliability, Security and Robustness in Heterogeneous Systems (QShine 2021)**, November 29-30, 2021*, vol. 402 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, pp. 159–173, Springer, 2021.

[7] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, "Device-to-device communications underlaying cellular networks," *IEEE Transactions on Communications*, vol. 61, pp. 3541–3551, August 2013.

[8] T. D. Hoang, L. B. Le, and T. Le-Ngoc, "Joint mode selection and resource allocation for relay-based d2d communications," *IEEE Communications Letters*, vol. 21, pp. 398–401, Feb 2017.

[9] C. Zhengwen, Z. Su, and S. Shixiang, "Research on relay selection in device-to-device communications based on maximum capacity," in *2014 International Conference on Information Science, Electronics and Electrical Engineering*, vol. 3, pp. 1429–1434, IEEE, April 2014.

[10] M. Liu and L. Zhang, "Joint power and channel allocation for relay-assisted device- to- device communications," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–5, IEEE, Aug 2018.

[11] X. Gu, M. Zhao, L. Ren, D. Wu, and S. Nie, "A two-stages relay selection and resource allocation with throughput balance scheme in relay-assisted d2d system," *Mobile Networks and Applications*, vol. 22, pp. 1020–1032, feb 2017.

[12] J. Deng, O. Tirkkonen, R. Freij-Hollanti, T. Chen, and N. Nikaein, "Resource allocation and interference management for opportunistic relaying in integrated mmWave/sub-6 GHz 5g networks," *IEEE Communications Magazine*, vol. 55, pp. 94–101, June 2017.

[13] Y. Zhao, Y. Li, X. Chen, and N. Ge, "Joint optimization of resource allocation and relay selection for network coding aided device-to-device communications," *IEEE Communications Letters*, vol. 19, pp. 807–810, May 2015.

[14] J. Sun, Z. Zhang, C. Xing, and H. Xiao, "Uplink resource allocation for relay-aided device-to-device communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, pp. 3883–3892, Dec 2018.

[15] R. N. Dutta and S. C. Ghosh, "Mobility aware resource allocation for millimeter-wave d2d communications in presence of obstacles," ***Computer Communications***,, vol. Vol. 200, pp. 54–65, feb 2023.