Spectrum Allocation: Algorithmic Approach

Shiv Prakash
Electrical Dept.
IIT Ropar
India
2021eeb1030@iitrpr.ac.in

Varun Sharma
Electrical Dept.
IIT Ropar
India
2021eeb1032@iitrpr.ac.in

Vivek Chadgal
Electrical Dept.
IIT Ropar
India
2021eeb1222@iitrpr.ac.in

Nipun Jugran
Electrical Dept.
IIT Ropar
India
2021eeb1305@iitrpr.ac.in

Dr. Satyam Agarwal
Electrical Dept.
IIT Ropar
India
satyam@iitrpr.ac.in

***Abstract*—Spectrum allocation is a critical aspect of modern telecommunications, ensuring efficient and equitable distribution of radio frequencies for various wireless communication services. Effective spectrum management has become paramount with the exponential growth of wireless technologies and the increasing demand for bandwidth-intensive applications. This paper provides an overview of spectrum allocation strategies, including regulatory frameworks, spectrum auctions, and spectrum-sharing techniques. It explores the challenges and opportunities associated with spectrum allocation, such as spectrum scarcity, interference mitigation, and emerging new wireless technologies like 5G and IoT. Furthermore, it discusses the role of governments, regulatory bodies, and industry stakeholders in shaping spectrum policies to promote innovation, competition, and socioeconomic development. By understanding the complexities of spectrum allocation, policymakers and industry players can collaborate to optimize spectrum utilization and foster the growth of wireless communications infrastructure worldwide.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Spectrum allocation refers to assigning portions of the electromagnetic spectrum to different services or users for various purposes, such as telecommunications, broadcasting, navigation, and scientific research. The electromagnetic spectrum encompasses all frequencies of electromagnetic radiation, from radio waves to gamma rays. Since the spectrum is a finite resource, its global allocation is managed by regulatory bodies. Allocation decisions are typically based on technical feasibility, efficient spectrum utilization, interference mitigation, and national or international regulations.

Algorithm's Requirements:

- Minimum interference between the channels.
- A minimum bandgap should be maintained.
- It should and must be fast and efficient.
- Some special channels need to be excluded.

A. Various approaches for designing algorithm

- **Frequency Division Multiple Access (FDMA):** Divides the frequency spectrum into multiple non-overlapping channels, and each channel is assigned to a specific user or communication service.
- **Geographic Spectrum Allocation:** Allocates specific frequency bands to different geographic regions to prevent interference between neighboring areas.
- **Frequency Reuse:** Divides a geographic area into smaller cells, and the same frequency band is reused in different cells at a sufficient distance to minimize interference.

B. Analysing the Dataset

This generates several points from the Poisson Distribution (Gaussian distribution). Because this distribution is found commonly in nature, we assume that the points will be distributed in this distribution. Then, it allocates coordinate values to the points generated using a uniform distribution process. The Code gives us the choice of Dense and Sparse spacing of Data Points generated by varying the Mean and variance of the distribution.

The Poisson distribution serves as a valuable tool in generating random datasets, particularly for modeling rare events occurring within a fixed interval of time or space. Its probability mass function (PMF) defines the likelihood of observing a specific number of events (k) given the average rate of occurrence (λ). By employing this distribution, we can simulate datasets wherein the frequency of events conforms to real-world scenarios, such as the number of calls received at a call center in an hour or the occurrence of emails in an inbox per day. To generate points from the Poisson distribution, we first establish the desired average rate of events (λ) and the dataset size (n). Subsequently, we employ the PMF to generate n random points, each representing the number of events at specific locations. These points are discrete and align with the characteristics of the Poisson distribution.

Allocating coordinate values to these points involves several steps. Initially, we assume a uniform distribution of points within a designated region, such as a square or a circle. To assign coordinates, we independently generate random values for the x and y coordinates from a uniform distribution within the predefined range, typically $[0, 1]$ for a unit square. Adjustments are made if the desired region differs from the unit square by scaling and shifting the coordinates accordingly. For instance, if the dataset should conform to a rectangle with dimensions $[a, b] \times [c, d]$, we map the uniformly generated coordinates to fit within this rectangle. Consequently, our final dataset comprises points distributed according to the Poisson distribution, with each point associated with specific coordinate values.

It's worth noting that due to the discrete nature of the Poisson distribution, the generated points will possess integer coordinates. For applications requiring continuous coordinates, alternative distributions like the Gaussian distribution or interpolation techniques may be more suitable. By adhering to these procedures, we can effectively simulate datasets with realistic event distributions, offering insights into various analytical and computational endeavors.

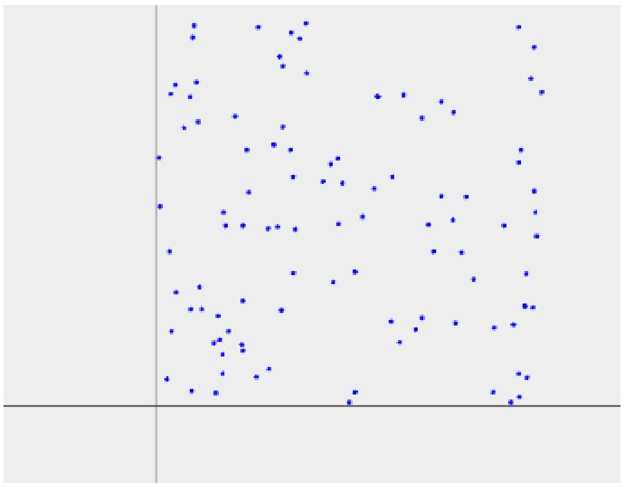

Fig. 1. Sparsely Distributed Data

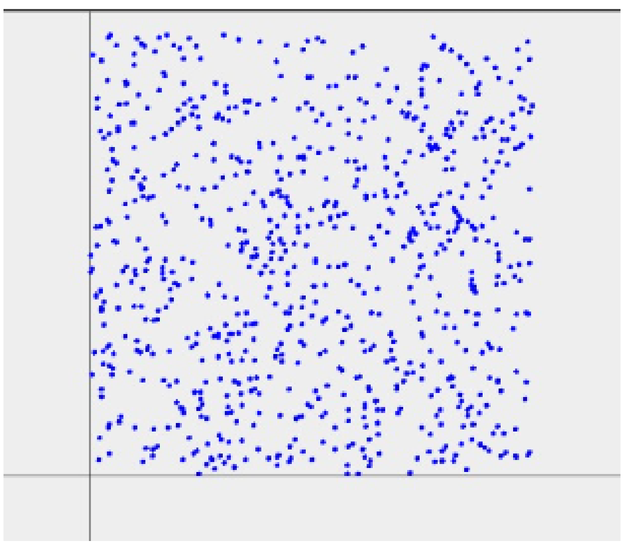

Fig. 2. Densely Distributed Data

II. Algorithmic Approach

This algorithm presents a method for allocating frequency channels in wireless sensor networks (WSNs) to mitigate interference and optimize network performance. By considering spatial proximity and interference constraints, the algorithm aims to assign channels to sensor nodes effectively.
Efficient channel allocation is vital for WSNs to ensure reliable communication and maximize network throughput. However, interference among sensor nodes operating on adjacent frequency channels can degrade performance. This algorithm addresses this challenge by proposing a novel channel allocation approach that accounts for spatial proximity and interference.

A. Objective Function

$$S = \sum_j D_j \cdot |F - F_j| \tag{1}$$

- F: Decision variable representing the channel to be assigned to the new location.
- F_j: Existing channel assignments at the jth location ($1 < j < n$).
- D_j: Weights representing the distance between the new location and the jth location. Higher D_j values indicate a greater impact on interference if F and F_j are close.

The objective function aims to minimize interference by maximizing the distance (absolute difference) between the new channel assignment F and existing channels at other locations F_j in the frequency domain . When F is far away from any existing F_j (large absolute difference), the interference between them is minimized. D_j weights emphasize the importance of avoiding interference with specific locations depending on the distance.

However, directly applying Jensen's inequality to the objective function S is challenging because it's a linear function of F. The key here is to reinterpret the weights D_j as a probability distribution P_j. This allows us to potentially leverage Jensen's inequality.

Reinterpreting Weights as Probabilities:

If we normalize the weights D_j such that they sum to 1:

$$[\frac{\sum D_j}{\sum(D_j \text{ for all } j)} = 1] \tag{2}$$

Then, the objective function can be rewritten as:

$$[S = \sum P(j) \cdot |(F - F_j)|] \tag{3}$$

where $P(j) = \frac{D_j}{\sum D_j}$ (represents the probability of considering the distance between the new location and the jth location).
If the function $f(x) = x$ is convex, then Jensen's inequality states:

$$[E[f(x)] \geq f(E[x])] \tag{4}$$

where $E[.]$ denotes the expectation.

In our case, $f(x) = x$ (the linear function representing the objective function). Let's analyze the change in the objective function (S) as F moves away from the average of the existing channel assignments ($E[F_j]$). We can do this by considering two cases:

Case 1: $F \gg E[F_j]$ or $F \ll E[F_j]$ (F is far away from the average): In this case, absolute differences $|(F - F_j)|$ will be larger. Let's denote the average positive difference as Δ_{far}:

$$[\Delta_{\text{far}} = \frac{1}{n} \sum |(F - F_j)|] \tag{5}$$

where n is the number of existing channel assignments. Then, the objective function S can be approximated as:

$$[S \approx \sum P(j) \cdot \Delta_{\text{far}}] \tag{6}$$

Since Δ_{far} is a positive constant and $P(j)$ is a probability distribution (summing to 1), the weighted sum S tends to be large when F is far away from the average.

Case 2: $F \approx E[F_j]$ (F is close to the average): In this case, the differences $|(F - F_j)|$ will be close to zero on average. We can denote the average difference as Δ_{near}:

$$[\Delta_{\text{near}} = \frac{1}{n} \sum |(F - F_j)|] \tag{7}$$

Then, the objective function S can be approximated as:

$$[S \approx \sum P(j) \cdot \Delta_{\text{near}}] \tag{8}$$

Here, Δ_{near} is generally smaller than Δ_{far} (from Case 1). This implies that the weighted sum S tends to be smaller when F is close to the average.

These cases suggest that the objective function S exhibits convexity. As F moves away from the average in either direction, the weighted sum S tends to increase, which aligns with the definition of a convex function.

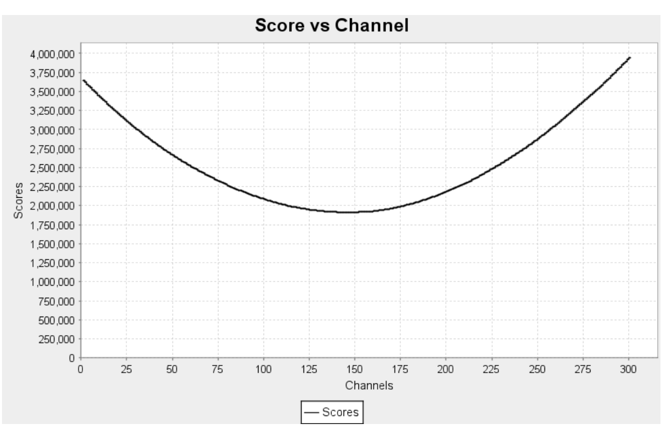

Fig. 3. Graph showing Score Convexity

Algorithm 1 Checking for Convexity

procedure CHECKCONVEXITY(scores : List of Doubles)

$l \leftarrow 0.5$ {Set lambda value}
isConvex $\leftarrow$ true {Assume convexity by default}
if size of scores ≤ 2 **then**
  Output "Can't check Convexity or concavity"
else
  for $i \leftarrow 1$ **to** size of scores $- 1$ **do**
    for $j \leftarrow 0$ **to** size of scores $- 1$ **do**
      {Apply modified Jensen's inequality for convex functions}
      if scores$[l \times (i + j)]$.getValue() $>$ $l \times$ scores$[i]$.getValue()$+ (1 - l) \times$ scores$[j]$.getValue() **then**
        isConvex $\leftarrow$ false
      end if
    end for
  end for
  if isConvex **then**
    Output "This is convex within range given"
  else
    Output "This is not convex within range given"
  end if
end if

B. Algorithm Description

The proposed algorithm begins prompting the user to input five integers: x_1, x_2, y_1, y_2, and d. These values define a rectangular region and a distance threshold used for subsequent calculations. Subsequently, the code reads data from multiple files. Each line in these files represents a signal, including its coordinates (X, Y), frequency channel, and a "closest coordinate" derived based on the defined region, to determine the closest coordinate within the specified rectangular region for a given point. The main loop iterates over a range of data sets, each containing coordinates and frequency channels obtained from the Dataset. For each data set, it calculates the squared distances between points and their closest coordinates, considering the transmission range. Based on these distances and the specified bandGap, it identifies candidate frequency channels that meet certain criteria, discarding those with potential interference.

Subsequently, the algorithm evaluates the spatial distribution of points and calculates scores for each frequency channel using Dynamic Programming (Prefix Sum Technique) to prioritize those with minimal interference and maximal coverage. These scores inform the selection of the final set of frequency channels, ensuring they are sufficiently spaced apart to minimize interference. The selection of the final set of frequency channel is done using the Two-Pointer technique based on the scores generated above. This facilitates the computational complexity as in this way we require only one iteration over the scores of all the channels based. Usage of this technique is possible due to the convexity of the score as discussed above.

Whereas, the Exhaustive Search algorithm initiates by setting parameters similar to the proposed code. For every dataset, it generates combinations of required frequency channels and assesses if the gap between any two channels in the combination surpasses the 'bandGap'. If this condition is met, it proceeds to validate if the selected combination adheres to interference avoidance criteria by assessing the distances between points and their closest coordinates. Upon detecting no interference, the algorithm computes the frequency distribution cost for each channel in the combination, considering its proximity to other points. These costs are then aggregated and stored alongside the corresponding combination. Upon processing all combinations, the algorithm identifies the combination with the minimal frequency distribution cost and presents its frequency channels as the optimal allocation solution for the provided dataset.

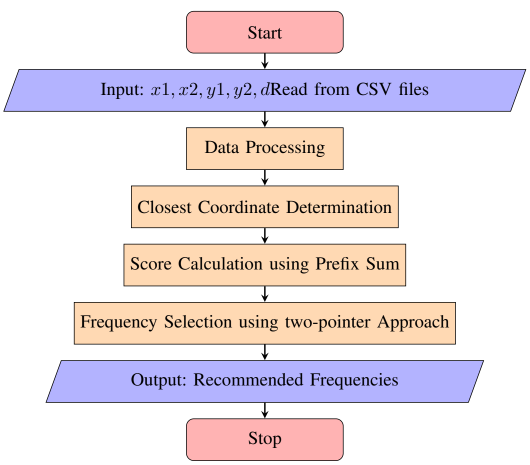

Fig. 4. Flowchart of the Proposed Algorithm

III. RESULTS

In the evaluation of frequency allocation spectrum, both the exhaustive search and the proposed prefix sum algorithm produced congruent solutions, showcasing the efficacy of the latter in achieving optimal frequency allocation. However, a crucial disparity emerged in their computational complexities. The exhaustive search method necessitates an exhaustive examination of all possible combinations in pairs, resulting in a time complexity that grows exponentially with the size of the spectrum. This exhaustive comparison process incurs substantial computational overhead, rendering it less scalable and efficient for larger spectrum sizes.

Contrastingly, the proposed prefix sum algorithm offers a more streamlined approach. By leveraging prefix sum calculations, it efficiently computes cumulative sums over the spectrum, thereby drastically reducing the number of comparisons required to identify the optimal frequency allocation. Consequently, the time complexity of the prefix sum algorithm remains significantly lower compared to the exhaustive search

Algorithm 2 Proposed Algorithm

procedure SELECTCHANNELS($x1, x2, y1, y2, d$)
$bandGap \leftarrow 5$
for $file\_name$ in range(14) **do**
  $candidates \leftarrow \{1, 2, ..., 300\}$
  for $type$ in ["dense", "sparse"] **do**
    Read CSV file for $type$
    for $line$ in CSV file **do**
      Calculate closest coordinates (cx, cy) and distance $dist$
      if $dist \leq (4 \times d + \text{line}[4])^2$ **then**
        Remove nearby channels from $candidates$
      end if
    end for
  end for
  Compute $score$ for each channel using Dynamic Programming (specifically Prefix Sum Technique)
  $back \leftarrow currentscoreofprefix$
  $front \leftarrow currentscoreofsuffix$
  $cur \leftarrow total$
  Append cur to $score$
  for i in range(1, 301) **do**
    $cur \leftarrow cur + back$
    $cur \leftarrow cur - front$
    for j in $distances[i]$ **do**
      $back \leftarrow back + j$
      $front \leftarrow front - j$
    end for
    Append cur to $score$
  end for
  Selection of channels based on their $scores$ using $two-pointer-approach$
  $reqFreq \leftarrow []$
  $req \leftarrow 10$
  $i \leftarrow 0$
  $j \leftarrow$ length of $candidates - 1$
  while req and $i \leq j$ **do**
    $f \leftarrow 0$
    if $score[candidates[i]] > score[candidates[j]]$ **then**
      $f \leftarrow candidates[i]$
      $i \leftarrow i + 1$

      $f \leftarrow candidates[j]$
      $j \leftarrow j - 1$
    end if
    if $f \neq 0$ **then**
      if f is not within $bandGap$ of any frequency in $reqFreq$ **then**
        Append f to $reqFreq$
        $req \leftarrow req - 1$
      end if
    end if
  end while
  Output $reqFreq$
end for

Algorithm 3 Exhaustive Search Technique for Comparision

procedure SELECTCHANNELS($x1, x2, y1, y2, d$)
$bandGap \leftarrow 5$
for $file\_name$ in range(3) **do**
  $dataSet \leftarrow$ Initialize empty dictionary
  $types \leftarrow$ ["new_dense", "new_sparse", "new_medium"]
  for $type$ in $types$ **do**
    Read CSV file for $type$
    for $line$ in CSV file **do**
      if line is not empty and line[0] is not "X" **then**
        Parse coordinates and other data
        Update $dataSet$
      end if
    end for
  end for
  $ans \leftarrow$ Empty list
  for $comb$ in combinations of frequencies **do**
    if valid combination according to $bandGap$ **then**
      $f \leftarrow$ True
      for k in $dataSet[comb[0]]$ **do**
        if distance condition not satisfied **then**
          $f \leftarrow$ False
          break
        end if
      end for{Similar checks for other frequencies}
      if f **then**
        Calculate frequency distances
        Append calculated distances to ans
      end if
    end if
  end for
  Sort ans based on distances
  Output the best combination from ans
end for

method, making it more suitable for real-world applications with sizable spectrum allocations.

In summary, while both methods deliver identical solutions, the proposed prefix sum algorithm demonstrates superior computational efficiency, presenting a promising avenue for practical implementation in spectrum allocation systems.

IV. CONCLUSION

In this project, we tackled the critical task of frequency allocation within a spectrum. Through meticulous analysis, we first established the convexity of the graph representing the frequency allocation problem. This foundational insight enabled us to develop and implement two distinct algorithms for frequency allocation: a conventional exhaustive search approach and a novel algorithm leveraging prefix sum calculations.

Our investigation revealed that the proposed algorithm exhibits a time complexity of O(n), affirming its superior efficiency compared to the traditional approach. By exploiting the convexity of the graph and employing innovative computational techniques, our algorithm streamlines the frequency allocation process, delivering optimal solutions in a fraction of the time required by traditional methods.

In conclusion, our findings underscore the importance of algorithmic innovation in addressing complex spectrum allocation challenges. The demonstrated superiority of our O(n) algorithm not only accelerates frequency allocation tasks but also lays the groundwork for more efficient spectrum management systems. Looking ahead, further research and development in this area hold the potential to revolutionize wireless communication networks, enhancing their capacity, reliability, and adaptability in the face of evolving demands.

Fig. 5. Complexity of Exhaustive Search Algorithm

Fig. 6. Complexity of Proposed Algorithm