

Bài tập:

1. Một hệ thống GSM có tổng số $T=600$ kênh tần số. Nếu sử dụng sơ đồ tái sử dụng tần số với $K=7$, hãy tính:

.

Ta có :

Tổng số kênh tần số: $T=600$

Hệ số tái sử dụng tần số: $K=7$

Số cụm cell trong hệ thống: $M=10$

a) Số kênh tần số mà mỗi cell có thể sử dụng

Trong một hệ thống GSM sử dụng sơ đồ tái sử dụng tần số với hệ số KKK, số kênh mà mỗi cell có thể sử dụng được tính bằng công thức:

$$C = T/K = 600/7 \approx 85.71$$

Vì số kênh phải là số nguyên, do đó có thể làm tròn thành 85 hoặc 86 tùy vào cách phân bổ thực tế.

b) Tổng dung lượng hệ thống nếu có $M= 10$ cụm cell.

Tổng số cell trong hệ thống được tính bằng:

$$N=M*K = 10*7 = 70$$

Dung lượng toàn hệ thống là:

$$D=N*C = 70 *8 = 5950 \text{ (nếu mỗi cell có 85 kênh)}$$

Hoặc

$$D=N*C = 70 *86 = 6020 \text{ (nếu mỗi cell có 86 kênh)}$$

Vậy:

Mỗi cell có thể sử dụng khoảng 85-86 kênh tần số.

Tổng dung lượng hệ thống khoảng 5950 - 6020 kênh tần số.

2. Giả sử một mạng di động sử dụng hệ số tái sử dụng $K=12$, với bán kính cell là $R=2\text{km}$. Hãy tính khoảng cách tối thiểu D giữa các cell đồng kênh.

Khoảng cách tối thiểu giữa các cell đồng kênh DDD được tính theo công thức:

$$D = \sqrt{3K} R = \sqrt{3 \cdot 12} \cdot 2 = 6 \cdot 2 = 12$$

Vậy khoảng cách tối thiểu giữa các cell đồng kênh là 12 km

3. Một hệ thống viễn thông có $K=19$ và $R=1.5\text{ km}$. Hãy tính khoảng cách tối thiểu giữa hai cell đồng kênh.

$$D = \sqrt{3K} R = \sqrt{57} \cdot 1,5 = 11.33$$

Vậy khoảng cách tối thiểu giữa hai cell đồng kênh là 11.33 km.

4. Một hệ thống di động có 106 thuê bao hoạt động đồng thời. Giả sử mỗi cuộc gọi chiếm 2 kênh và tổng số kênh trong hệ thống là 5000, hãy tính:

a) Số cụm cell cần thiết để phục vụ toàn bộ hệ thống nếu $K=7$.

b) Tổng dung lượng của hệ thống

a,

Bước 1: Xác định số kênh cần thiết cho các thuê bao

Mỗi thuê bao cần 2 kênh, nên tổng số kênh cần thiết để phục vụ 106 thuê bao là:

$$\text{Số kênh cần thiết} = 106 \cdot 2 = 212$$

Bước 2: Tính số kênh trên mỗi cụm cell

Số kênh trong một cụm cell được tính bằng:

Số kênh trên một cum cell = Tổng số kênh / K = 5000 / 7 ≈ 714.3 kênh/cum cell

Bước 3: Xác định số cum cell cần thiết

Số cum cell cần thiết để phục vụ tất cả thuê bao là:

$$\text{Số kênh cần thiết} / \text{Số kênh trên một cum cell} = 212 / 714.3 \approx 0.297$$

Vì số cum cell phải là một số nguyên, ta làm tròn lên 1 cum cell, đủ để phục vụ thuê bao

b,

Tổng dung lượng hệ thống được tính bằng tổng số cuộc gọi mà hệ thống có thể hỗ trợ cùng lúc:

$$\text{Tổng số kênh hệ thống} / \text{Số kênh mỗi cuộc gọi} = 5000/2 = 2500$$

⇒ Số cum cell cần thiết: 1 cum cell

Tổng dung lượng hệ thống: 2500 cuộc gọi đồng thời

Bài tập (Tình huống thực tế):

1. Quy hoạch mạng di động:

Anh là một kỹ sư tư vấn di động cho một thành phố mới. Anh/Chị được giao nhiệm vụ lựa chọn hệ số K phù hợp.

1. Bạn sẽ chọn K=7, K=12 hay K=10?

2. Lựa chọn của Anh/Chị dựa trên những yếu tố nào?

Tôi sẽ chọn K = 7 .

Vì :

K=7:

Cung cấp nhiều kênh tần số hơn cho mỗi cell → Dung lượng cao

Khoảng cách giữa các cell đồng kênh ngắn hơn → Gây nhiễu đồng kênh cao hơn

Phù hợp cho khu vực đô thị đông dân, cần nhiều dung lượng.

2. Giảm nhiễu đồng kênh: Một thành phố đang gặp vấn đề về nhiễu đồng kênh nghiêm trọng trong mạng GSM hiện tại. Hãy đề xuất 3 giải pháp giúp cải thiện vấn đề này.

1. Tăng hệ số tái sử dụng tần số (K)

Cách thực hiện:

Hiện tại, nếu K thấp (ví dụ: $K=7$), khoảng cách giữa các cell đồng kênh ngắn, gây nhiễu mạnh.

Tăng K lên (ví dụ: từ $K=7$ lên $K=10$ hoặc $K=12$) giúp cell đồng kênh xa nhau hơn, giảm nhiễu.

2. Điều chỉnh công suất phát của trạm BTS

Cách thực hiện:

Giảm công suất phát của trạm gốc (BTS) giúp giảm phạm vi phủ sóng, hạn chế can nhiễu đến cell lân cận dùng chung tần số.

Cấu hình Power Control để BTS tự động điều chỉnh công suất phù hợp với điều kiện thực tế.

3. Sử dụng kỹ thuật ăng-ten thông minh (Smart Antennas) hoặc Ăng-ten định hướng

Cách thực hiện:

Thay vì dùng ăng-ten vô hướng (Omni), chuyển sang ăng-ten định hướng (Sector Antennas) để chia mỗi cell thành 3 hoặc 6 sector.

Điều chỉnh góc phủ sóng và độ nghiêng (tilt) để tránh tín hiệu lan quá xa gây nhiễu.

3. Mạng LTE ở vùng nông thôn: Mạng LTE được triển khai ở một vùng nông thôn rộng lớn. Bạn có nghĩ rằng hệ số $K=3$ là phù hợp không? Tại sao?

Với $K=3$:

Mỗi cụm cell chỉ có 3 tần số khác nhau, tức là tần số được tái sử dụng sau mỗi 3 cell.

Điều này giúp tối ưu dung lượng, vì mỗi cell có nhiều băng tần hơn.

Tuy nhiên, khoảng cách giữa các cell đồng kênh sẽ nhỏ hơn, có thể gây nhiễu đồng kênh nếu không kiểm soát tốt.

Vùng nông thôn có những đặc điểm riêng biệt:

Mật độ người dùng thấp → Không cần quá nhiều kênh tần số trên mỗi cell.

Diện tích phủ sóng lớn → Cần khoảng cách xa hơn giữa các cell đồng kênh để giảm nhiễu.

Ít vật cản (tòa nhà, công trình) → Sóng truyền xa hơn, dễ gây nhiễu giữa các cell.

→ Với điều kiện này, chọn $K=3$ có thể không phù hợp do nhiễu đồng kênh cao.

Bài tập thực hành (Mô phỏng)

Các bài tập thực hành có thể được triển khai bằng Python hoặc phần mềm mô phỏng mạng.

1. Mô phỏng quy hoạch tần số

- Viết chương trình Python để hiển thị sơ đồ cell với các giá trị $K=3, 7, 12$.

- Mô phỏng cách xác định vị trí của các cell đồng kênh bằng cách sử dụng công thức

a,

```
import numpy as np
import matplotlib.pyplot as plt

def hexagon(center, size=1, color='b'):
    """Vẽ một cell hình lục giác với tâm và kích thước cho trước"""
    angles = np.linspace(0, 2 * np.pi, 7)
    x = center[0] + size * np.cos(angles)
    y = center[1] + size * np.sin(angles)
    plt.fill(x, y, color=color, alpha=0.5, edgecolor='k')

def generate_cells(K, grid_size=3, cell_size=1): # Giảm kích thước lưới
    """Tạo và hiển thị sơ đồ cell với hệ số tái sử dụng K"""
    plt.figure(figsize=(6, 6)) # Thu nhỏ biểu đồ
    colors = plt.cm.get_cmap('tab10', K) # Chọn bảng màu cho các nhóm K

    # Tạo lưới hình lục giác
    for i in range(-grid_size, grid_size + 1):
        for j in range(-grid_size, grid_size + 1):
            x = cell_size * (1.5 * i)
            y = cell_size * (np.sqrt(3) * (j + 0.5 * (i % 2)))
            group = (i * j) % K # Xác định nhóm tần số
            hexagon((x, y), cell_size, color=colors(group))
            plt.text(x, y, str(group), ha='center', va='center', fontsize=8, color='black')

    plt.xlim(-grid_size * 2, grid_size * 2)
    plt.ylim(-grid_size * 2, grid_size * 2)
    plt.axis('off')
    plt.title(f"Sơ đồ quy hoạch tần số với  $K=\{K\}$ ")
```

```
plt.show()
```

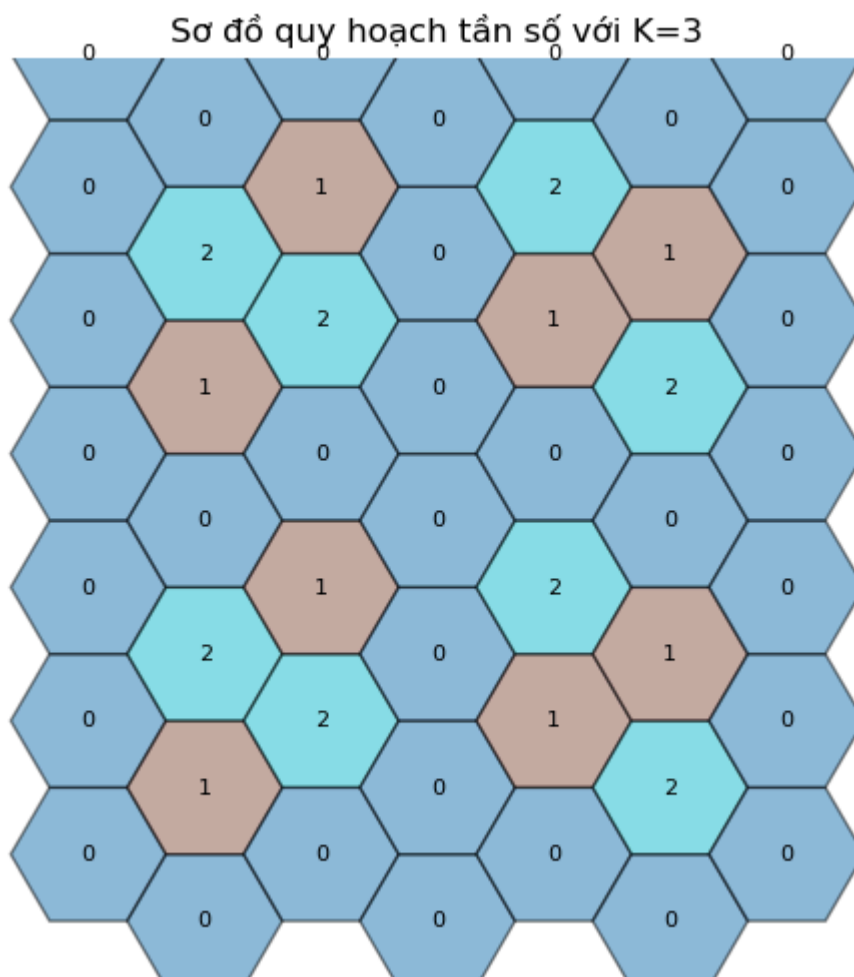
Chạy mô phỏng với các giá trị K

```
generate_cells(K=3)
```

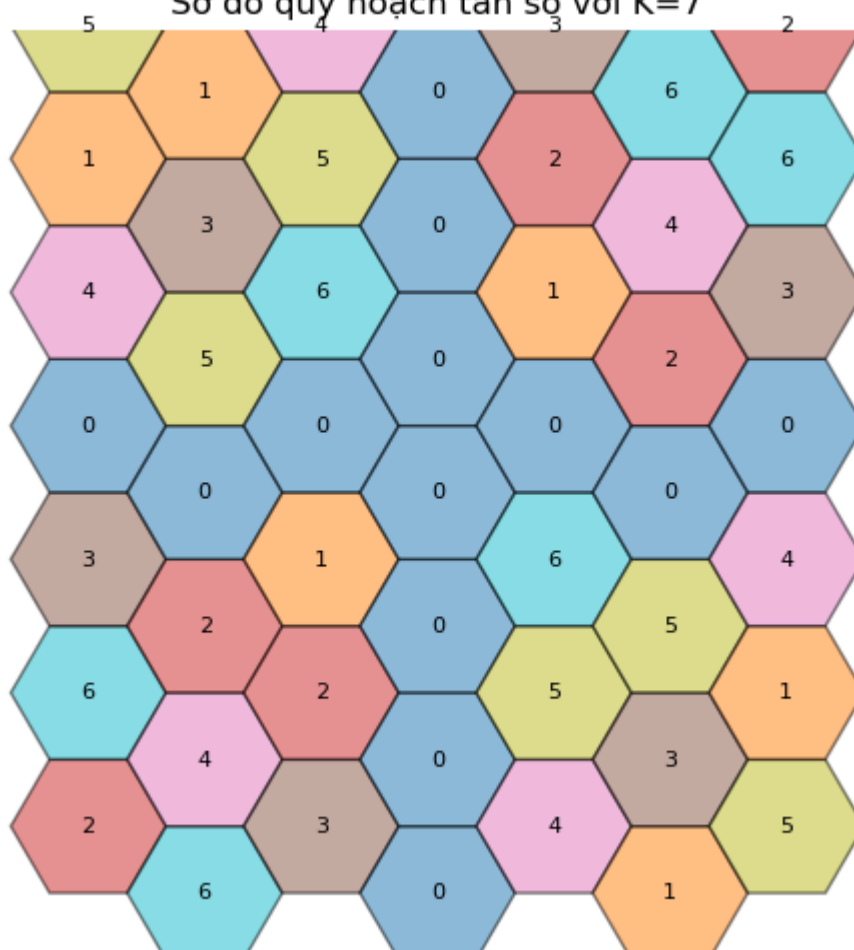
```
generate_cells(K=7)
```

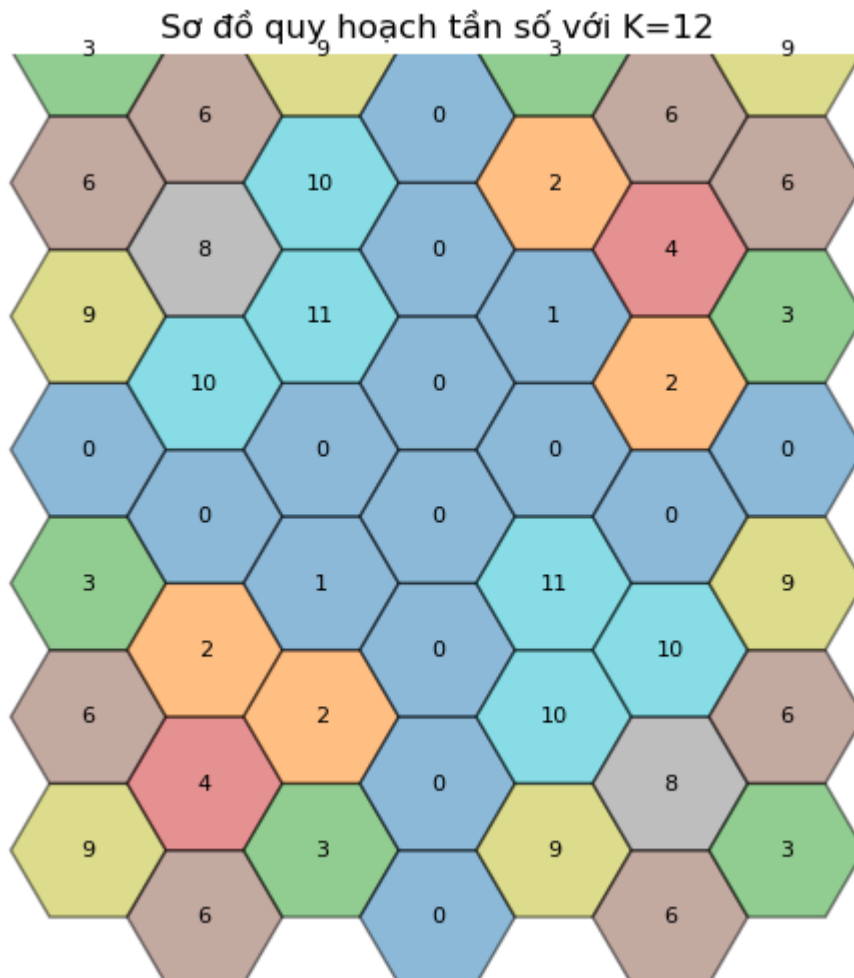
```
generate_cells(K=12)
```

Kết quả :



Sơ đồ quy hoạch tần số với K=7





b,

Để mô phỏng vị trí của các cell đồng kênh, chúng ta sử dụng công thức tính khoảng cách giữa các cell đồng kênh:

```
import numpy as np
import matplotlib.pyplot as plt

def hexagon(center, size=1, color='b', label=None):
    """Vẽ một cell hình lục giác với tâm và kích thước cho trước"""
```

```

angles = np.linspace(0, 2 * np.pi, 7)
x = center[0] + size * np.cos(angles)
y = center[1] + size * np.sin(angles)
plt.fill(x, y, color=color, alpha=0.5, edgecolor='k')
if label is not None:
    plt.text(center[0], center[1], label, ha='center', va='center', fontsize=8,
color='black')

def generate_cells(K, grid_size=3, cell_size=1):
    """Tạo và hiển thị sơ đồ cell với hệ số tái sử dụng K và đánh dấu cell đồng
kênh"""
    plt.figure(figsize=(6, 6))
    colors = plt.cm.get_cmap('tab10', K) # Chọn bảng màu cho các nhóm K

    cell_positions = [] # Lưu vị trí các cell

    # Tạo lưới hình lục giác
    for i in range(-grid_size, grid_size + 1):
        for j in range(-grid_size, grid_size + 1):
            x = cell_size * (1.5 * i)
            y = cell_size * (np.sqrt(3) * (j + 0.5 * (i % 2)))
            group = (i * j) % K # Xác định nhóm tần số
            hexagon((x, y), cell_size, color=colors(group), label=str(group))
            cell_positions.append(((x, y), group))

    # Tính khoảng cách đồng kênh theo công thức
    D = np.sqrt(3 * K) * cell_size
    plt.title(f"Sơ đồ quy hoạch tần số với K={K} (D = {D:.2f})")

    plt.xlim(-grid_size * 2, grid_size * 2)

```

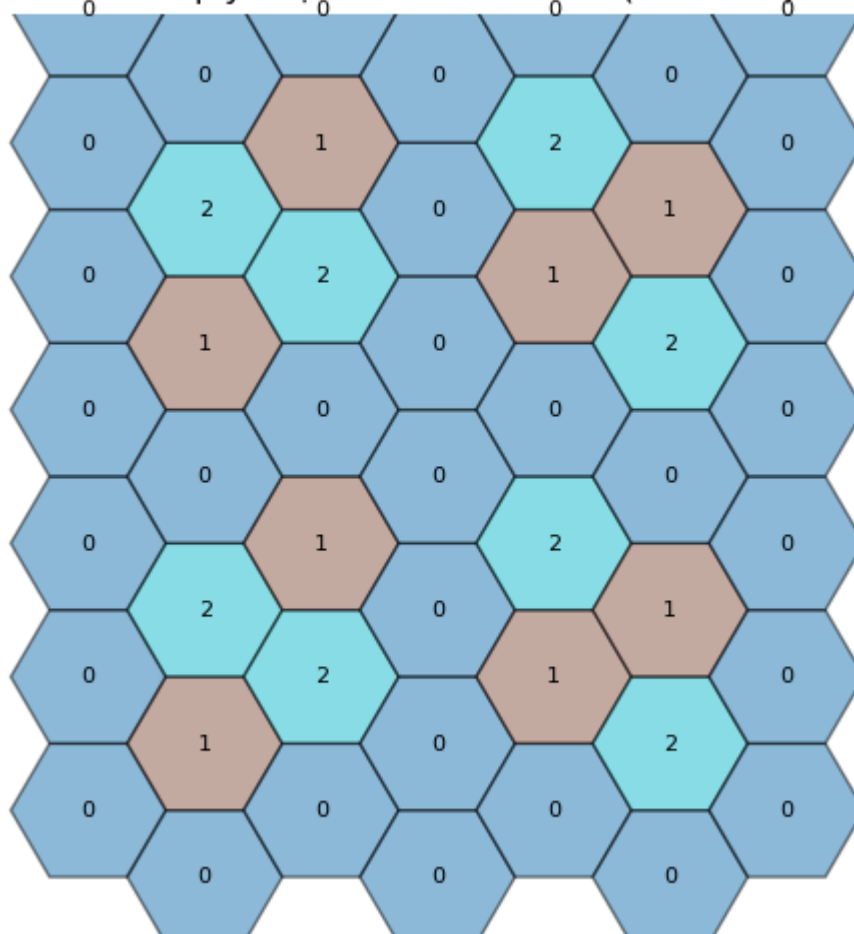
```
plt.ylim(-grid_size * 2, grid_size * 2)
plt.axis('off')
plt.show()

return cell_positions, D

# Chạy mô phỏng với các giá trị K
for K in [3, 7, 12]:
    cell_positions, D = generate_cells(K)
    print(f"Khoảng cách đồng kênh D cho K={K}: {D:.2f}")
```

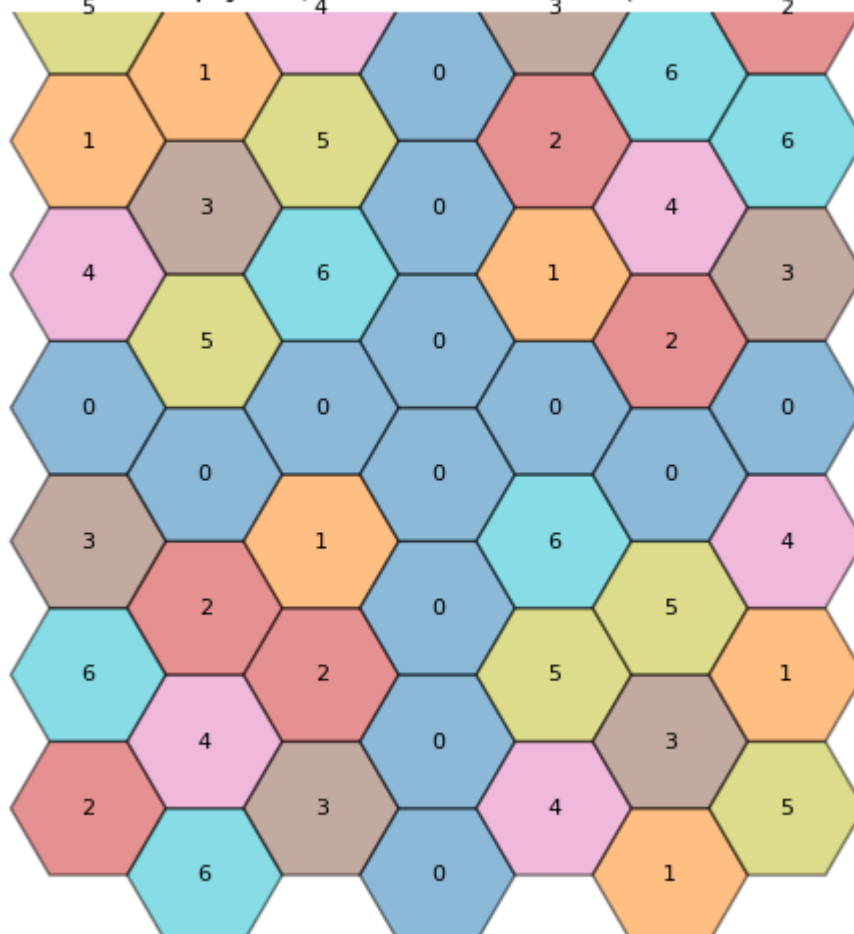
Kết quả :

Sơ đồ quy hoạch tần số với $K=3$ ($D = 3.00$)

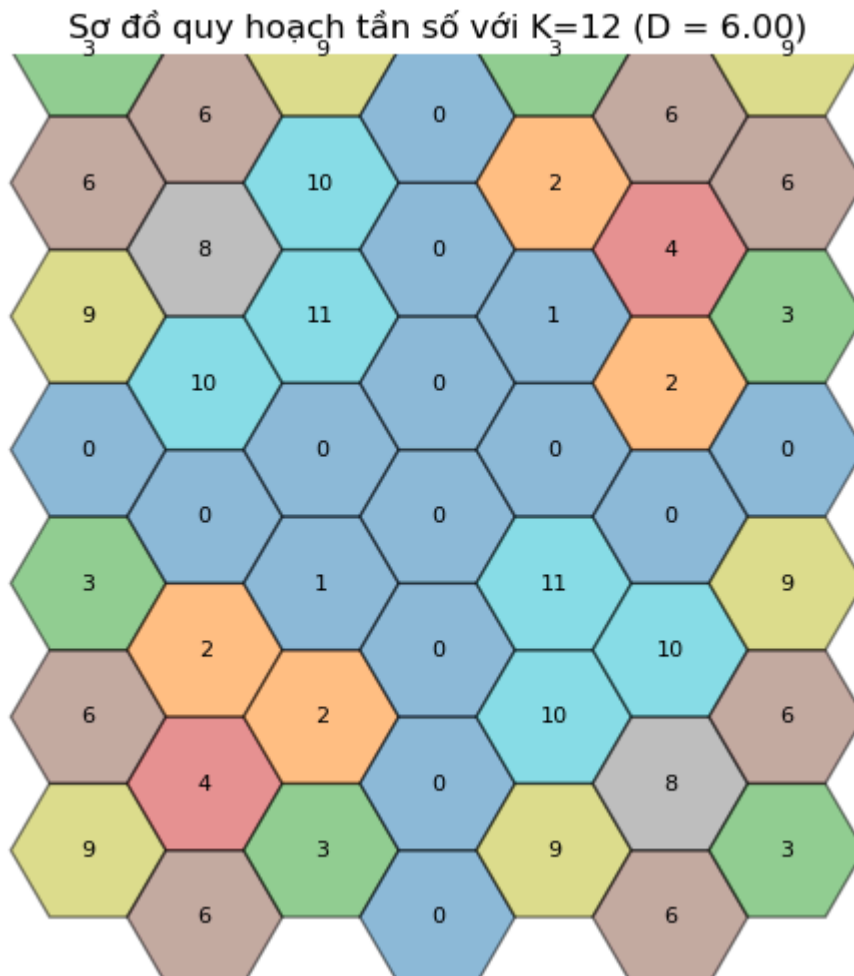


Khoảng cách đồng kênh D cho $K=3$: 3.00

Sơ đồ quy hoạch tần số với $K=7$ ($D = 4.58$)



Khoảng cách đồng kênh D cho $K=7$: 4.58



Khoảng cách đồng kênh D cho $K=12$: 6.00

2 Tính toán nhiễu đồng kênh bằng Python

- Viết một chương trình để tính toán mức nhiễu đồng kênh dựa trên khoảng cách tái sử dụng D.
- So sánh nhiễu giữa các giá trị $K=7$, $K=12$.

a,

```
import numpy as np
```

```

import matplotlib.pyplot as plt

def hexagon(center, size=1, color='b', label=None):
    """Vẽ một cell hình lục giác với tâm và kích thước cho trước"""
    angles = np.linspace(0, 2 * np.pi, 7)
    x = center[0] + size * np.cos(angles)
    y = center[1] + size * np.sin(angles)
    plt.fill(x, y, color=color, alpha=0.5, edgecolor='k')
    if label is not None:
        plt.text(center[0], center[1], label, ha='center', va='center', fontsize=8,
color='black')

def generate_cells(K, grid_size=3, cell_size=1):
    """Tạo và hiển thị sơ đồ cell với hệ số tái sử dụng K và đánh dấu cell đồng
kênh"""
    plt.figure(figsize=(6, 6))
    colors = plt.cm.get_cmap('tab10', K) # Chọn bảng màu cho các nhóm K

    cell_positions = [] # Lưu vị trí các cell

    # Tạo lưới hình lục giác
    for i in range(-grid_size, grid_size + 1):
        for j in range(-grid_size, grid_size + 1):
            x = cell_size * (1.5 * i)
            y = cell_size * (np.sqrt(3) * (j + 0.5 * (i % 2)))
            group = (i * j) % K # Xác định nhóm tần số
            hexagon((x, y), cell_size, color=colors(group), label=str(group))
            cell_positions.append(((x, y), group))

    # Tính khoảng cách đồng kênh theo công thức

```

```

D = np.sqrt(3 * K) * cell_size
plt.title(f"Sơ đồ quy hoạch tần số với K={K} (D = {D:.2f})")

plt.xlim(-grid_size * 2, grid_size * 2)
plt.ylim(-grid_size * 2, grid_size * 2)
plt.axis('off')
plt.show()

return cell_positions, D

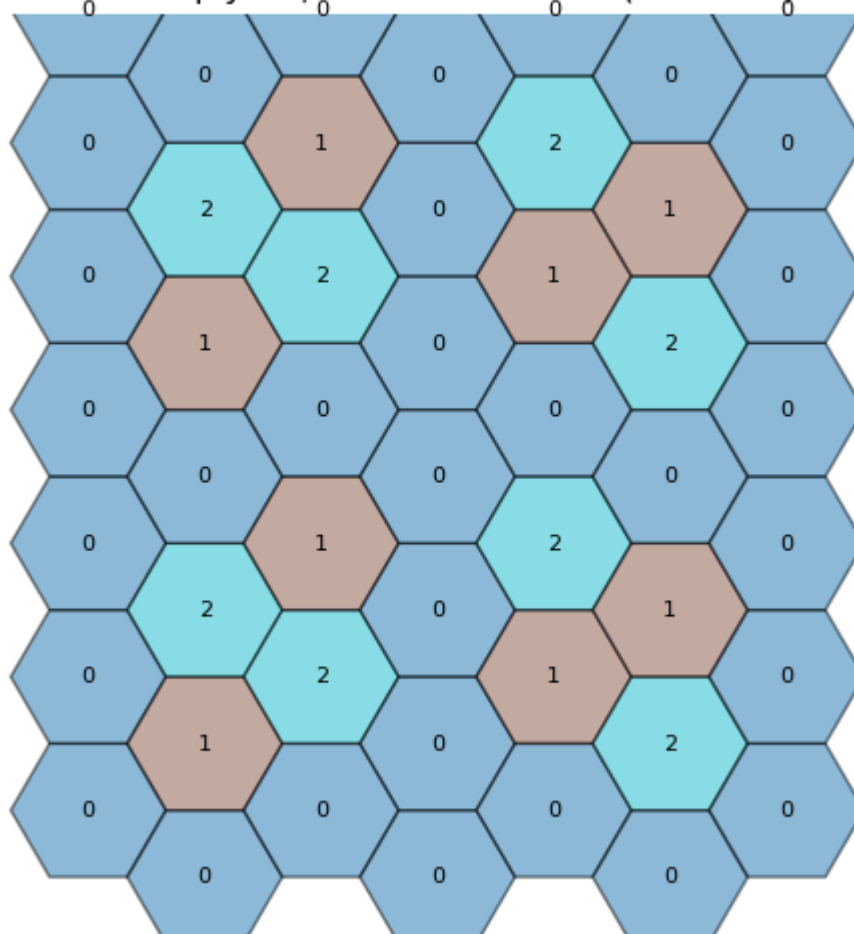
def calculate_cochannel_interference(K, n=4):
    """Tính toán mức nhiễu đồng kênh dựa trên khoảng cách D"""
    D = np.sqrt(3 * K) # Khoảng cách đồng kênh
    N = 6 # Số cell đồng kênh gần nhất
    interference = sum([1 / (D ** n) for _ in range(N)])
    SIR = 1 / interference # Tỷ số tín hiệu trên nhiễu (S/I)
    print(f"Mức nhiễu đồng kênh (S/I) cho K={K}: {SIR:.2f} dB")
    return SIR

# Chạy mô phỏng với các giá trị K
for K in [3, 7, 12]:
    cell_positions, D = generate_cells(K)
    print(f"Khoảng cách đồng kênh D cho K={K}: {D:.2f}")
    calculate_cochannel_interference(K)

```

Kết quả :

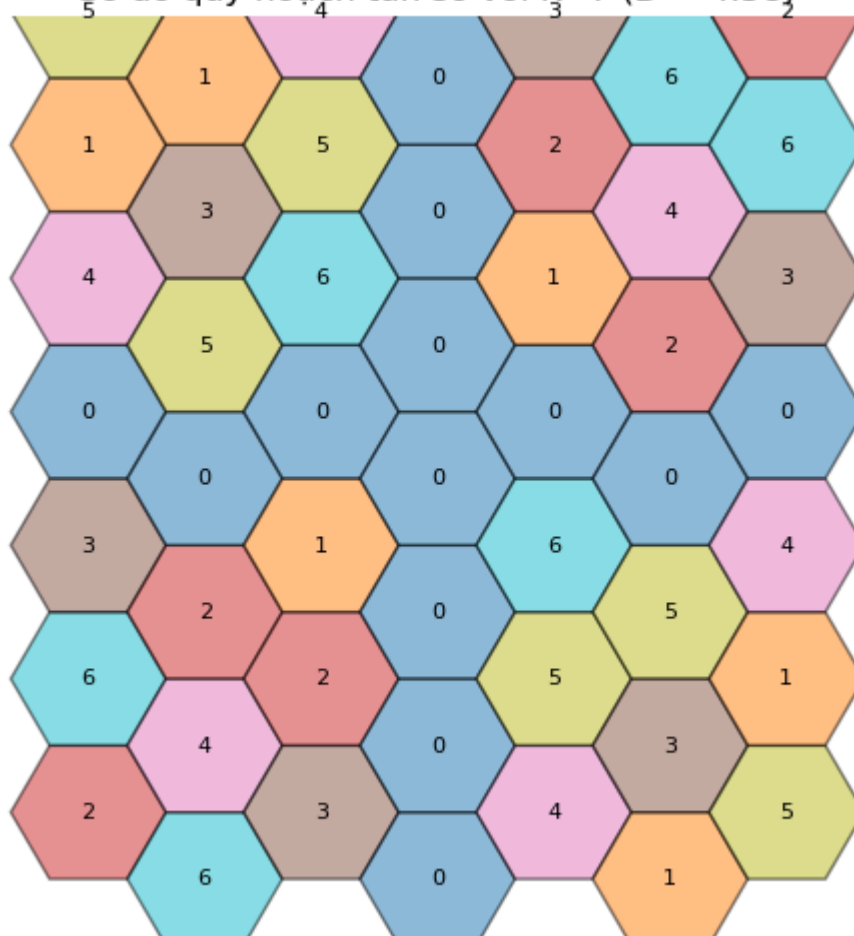
Sơ đồ quy hoạch tần số với $K=3$ ($D = 3.00$)



Khoảng cách đồng kênh D cho $K=3$: 3.00

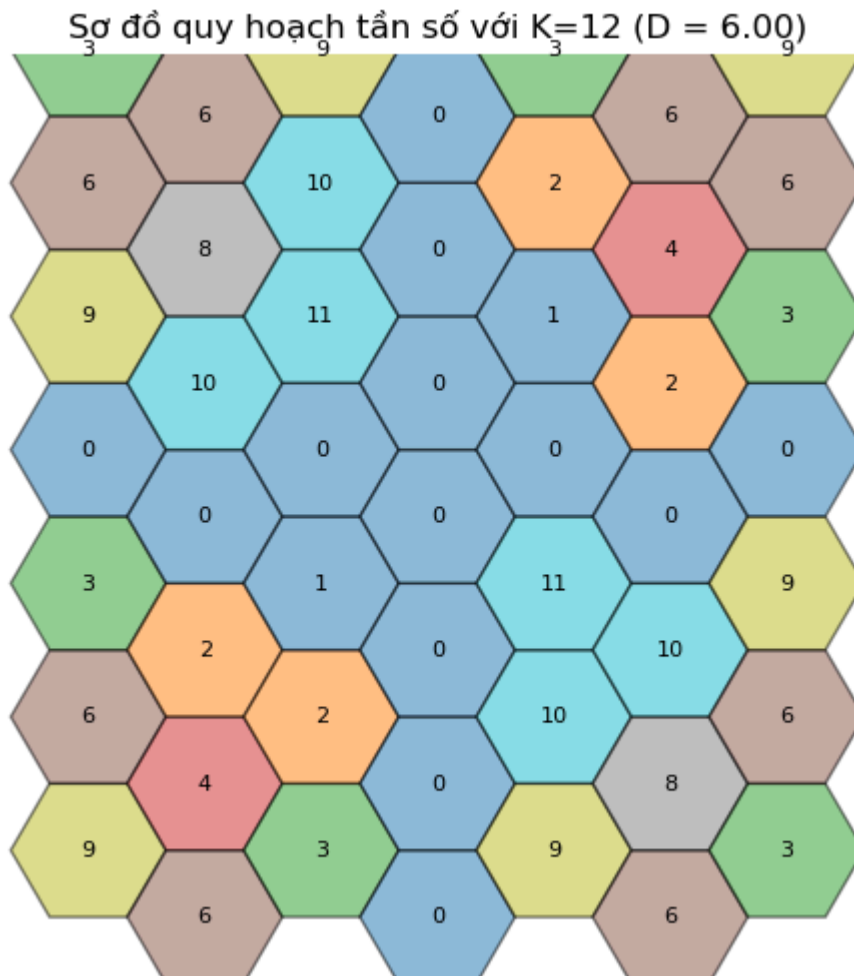
Mức nhiễu đồng kênh (S/I) cho $K=3$: 13.50 dB

Sơ đồ quy hoạch tần số với $K=7$ ($D = 4.58$)



Khoảng cách đồng kênh D cho $K=7$: 4.58

Mức nhiễu đồng kênh (S/I) cho $K=7$: 73.50 dB



Khoảng cách đồng kênh D cho $K=12$: 6.00

Mức nhiễu đồng kênh (S/I) cho $K=12$: 216.00 dB

b,

Khoảng cách đồng kênh (D)

Với $K=7$: $D=4.58$

Với $K=12$: $D=6.000$

Khoảng cách D càng lớn, mức nhiễu đồng kênh càng giảm.

Với $K=7$, D nhỏ hơn, dẫn đến mức nhiễu đồng kênh cao hơn.

Với $K=12$, D lớn hơn, giúp giảm nhiễu đáng kể.

