WIRELESS PROJECT

DR/MICHAEL IBRAHIM
ENG/ NADA NADER

Name	ID
Mohamed Mostafa Elnahas Ibrahim	1901553
Youssef Eslam Mohamed Mokhtar	1901223
Tarek Magdy Sayed Taha	1900702
Rama Ayman Mohamed	1901111

Table of Contents

Part A

1) Calculate Cluster Size

$$SIR_{Ratio} = 10^{\frac{SIR_{dB}}{10}}$$

$$Q = \frac{D}{R} = \sqrt{3N}$$

$$SIR_{Ratio} = \frac{\left(\sqrt{3N} - 1\right)^n}{i_o}$$

$$N \geq \frac{1}{3}\left(\sqrt[n]{i_o * 10^{\frac{SIR_{Ratio}}{10}}} + 1\right)^2$$

Where:

SIR: Signal-to-interference ratio (dB).

Io: Number of interfering co-channels.

For $i_o = 1$ $using\ 60^o\ Sectorization$

For $i_o = 2$ $using\ 120^o\ Sectorization$

For $i_o = 6$ $using\ omni\ directional$

n: path loss exponent.

N: Cluster Size.

$$N = i^2 + ik + k^2$$

N=1,3,4,7,9,13,16,19,21,25,27,28,31,36,37,39,43,48,49

2) Number of Cells

Total $number\ of\ Channels\ per\ sector = \left\lfloor \frac{S}{N * number\ of\ sectors} \right\rfloor$

$$P_r = \frac{\frac{A^c}{C!}}{\sum_{K=0}^{c} \frac{A^k}{K!}} = GOS$$

Where P_r *(GOS) is the probability of Blocking Calls "Erlang-B".*

$$A = Capacity * A_u$$

$$A = User\ Density\ *\ City\ Area * A_u$$

$$A\ per\ cell = A\ per\ sector\ *\ number\ of\ sectors$$

$$Total\ Number\ of\ cells\ = \left\lceil \frac{A}{A\ Per\ Cell} \right\rceil$$

Where A is traffic intensity

3) Cell Radius

$$Cell\ Area\ = \frac{City\ Area}{Number\ of\ Cells}\ = \frac{3\sqrt{3}}{2}\ R^2$$

$$Cell\ Radius\ (R) = \sqrt{\frac{Cell\ Area}{\frac{3\sqrt{3}}{2}}}$$

4) Calculation of Base Station Transmitted Power

Using hata model for urban medium-sized city

$$L_U\ =\ 69.55\ +\ 26.16\ log_{10} f\ -\ 13.82\ log_{10} h_{Tx} - C_H +\ (44.9\ -\ 6.55\ log10\ h_{Tx})\ *\ log_{10} d$$

$$C_H\ =\ 0.8\ +\ (1.1\ log_{10} f - 0.7)\ \ h_{Rx}\ -\ 1.56\ (log_{10} f)$$

Where:

- h_{Rx} *: mobile station antenna heigh (m).*
- h_{Tx}*: base station antenna height of (m).*
- C_H*: Antenna height correction factor.*
- f*: Frequency (MHz).*
- L_U*: Path loss in urban areas (dB).*
- d*: Distance between base station and mobile station (Km).*
- *If Mobile Station is at the edge of the cell, the distance is R.*

Received power calculated with relation: $P_{Rx}\ \ = P_{Tx} -\ Path\ loss$

5) MATLAB code for part 1:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%Part 1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
clc;
close all;
%Constants
channels = 340;
f = 900; % Frequency in MHz *10^9
hb = 20; % BS height in meters
hm = 1.5; % MS height in meters
sensitivity = -95; % MS sensitivity in dBm
traffic_per_user = 0.025; % Traffic intensity per user in Erlangs
path_loss_exponent = 4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ask user for input parameters
GOS = input('GOS (ex:0.02): ');
city_area = input('The City Area (In km^2): ');
user_density = input('User Density (users/km^2): ');
SIRmin_dB = input('Minimum SIR Required (In dB): ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Choosing The Method Of Sectorization
input1 = 1;
while input1
    sectorization_method = input(['Please Enter The Sectorization method: ...' ...
        '\n a = omni-directional \n b = 120° sectorization  \n c = 60° sectorization \n
'], 's');
    sectorization_method = lower(sectorization_method); % Convert input to lowercase
    if sectorization_method == 'a'
        sectors = 1;
        i0 = 6;
        input1 = 0;
    elseif sectorization_method == 'b'
        sectors = 3;
        i0 = 2;
        input1 = 0;
    elseif sectorization_method == 'c'
        sectors = 6;
        i0 = 1;
        input1 = 0;
    else
        disp('Wrong parameter')
        input1 = 1;
    end
end
fprintf('\n');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Cluster Size Calculations
SIR_ratio = 10^(SIRmin_dB/10); % SIR ratio in dB
Q = (SIR_ratio*i0)^(1/path_loss_exponent);
A_TEMP = ((Q+1).^2)/3;
Cluster_size = ceil(A_TEMP);
found_n = false;  % flag to indicate if a solution has been found
while ~found_n
    for i = 0:Cluster_size
        for k = 0:Cluster_size
            if i^2 + i*k + k^2 == Cluster_size
                found_n = true;
```

```
                break;
            end
        end
        if found_n
            break;
        end
    end
    if ~found_n
        Cluster_size = Cluster_size + 1;
    end
end
disp(['Cluster_size = ', num2str(Cluster_size)]);%->1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculating Intensity and Number of cells

num_channels_per_sector = floor(channels / (Cluster_size * sectors));

%Solving The Erling B equation using fzero function

fun = @(A) GOS - (A^num_channels_per_sector/factorial(num_channels_per_sector)) ...
/ sum(A.^((0:num_channels_per_sector))./factorial(0:num_channels_per_sector));
traffic_intensity_per_sector = fzero(fun, [0, 1000]);

traffic_intensity_per_cell = traffic_intensity_per_sector * sectors;
total_traffic_intensity = user_density * city_area * traffic_per_user;
Num_cells = ceil((total_traffic_intensity) / traffic_intensity_per_cell);
disp(['Total Number of Cells = ', num2str(Num_cells)]);%->2
disp(['Traffic Intensity per Cell = ', num2str(traffic_intensity_per_cell), '
Erlang']);%->4
disp(['Traffic Intensity per Sector = ', num2str(traffic_intensity_per_sector), '
Erlang']);%->5
cell_Area = city_area/Num_cells;
cell_red = sqrt((2*cell_Area)/(3*sqrt(3)));
disp(['Cell radius = ', num2str(cell_red),' Km']);%->3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the transmitted power
L= Hata(f, hm, hb, cell_red);
Ptx = sensitivity+ L ;
disp(['Ptx = ', num2str(Ptx),' dBm']);%->6
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot the MS received power as a function of distance
D=Q*cell_red;
d = 0:0.01:D;
L= Hata(f, hm, hb, d);
Prx = Ptx - L ;
plot(d, Prx, 'LineWidth', 2);%->7
xlabel('Distance from BS (km)');
ylabel('Received Power (dBm)');
title('Received Power vs. Distance');
grid on
```

Comments:

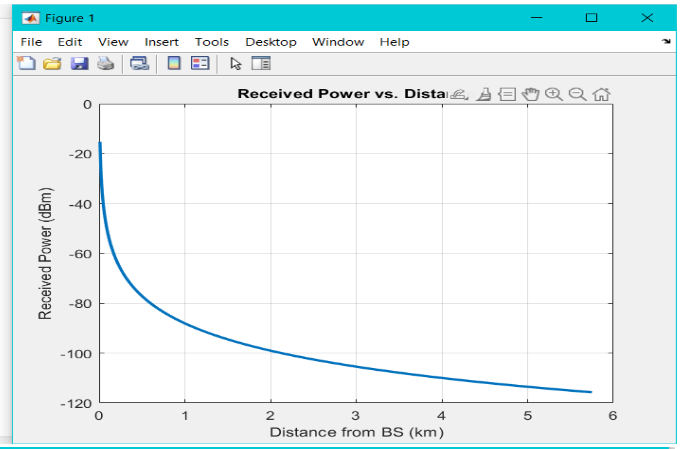
The relation between the distance and received power is an inverse proportional, which means that as the distance from the base station increases, the received power at the mobile station decreases.

This relation is derived from the Hata model used in the code, which considers the path loss due to propagation in the environment. The path loss increases with increasing distance, resulting in a decrease in the received power at the mobile station.

6) *Examples Graph:*

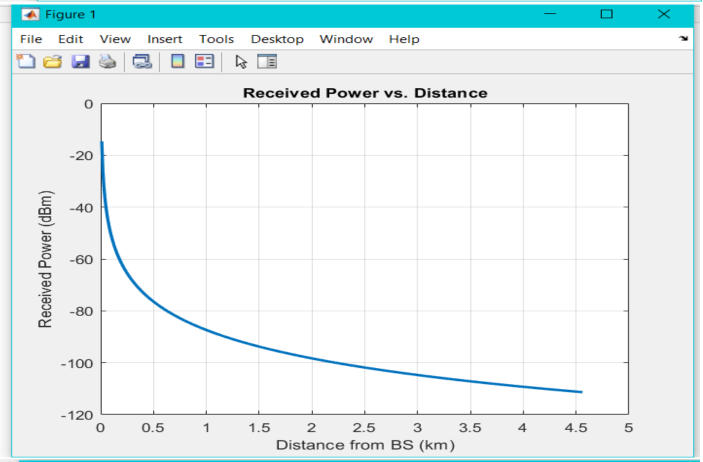
Command Window

```
GOS (ex:0.02): 0.05
The City Area (In km^2): 300
User Density (users/km^2): 200
Minimum SIR Required (In dB): 15
Please Enter The Sectorization method:
 a = omni-directional
 b = 120° sectorization
 c = 60° sectorization
 a

Cluster_size = 9
Total Number of Cells = 48
Traffic Intensity per Cell = 31.6397 Erlang
Traffic Intensity per Sector = 31.6397 Erlang
Cell radius = 1.551 Km
Ptx = 40.5938 dBm
fx >>
```



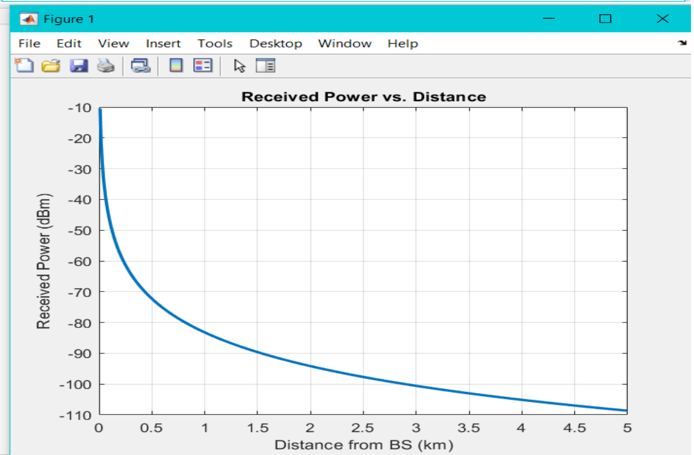
Command Window

```
GOS (ex:0.02): 0.05
The City Area (In km^2): 300
User Density (users/km^2): 200
Minimum SIR Required (In dB): 15
Please Enter The Sectorization method:
 a = omni-directional
 b = 120° sectorization
 c = 60° sectorization
 b

Cluster_size = 7
Total Number of Cells = 44
Traffic Intensity per Cell = 34.6308 Erlang
Traffic Intensity per Sector = 11.5436 Erlang
Cell radius = 1.62 Km
Ptx = 41.2812 dBm
fx >>
```



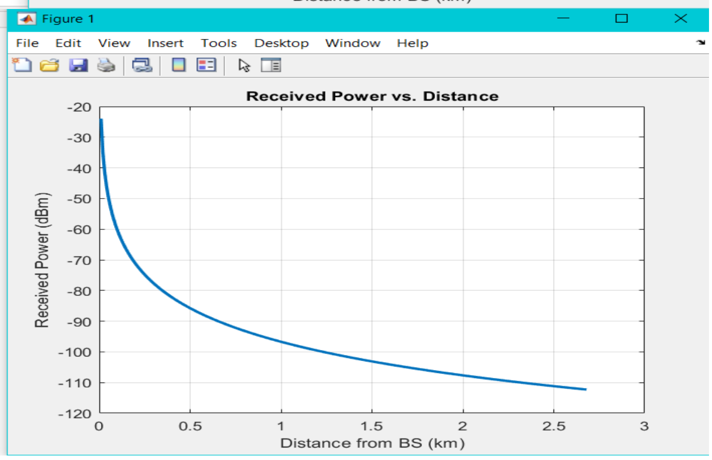
Command Window

```
GOS (ex:0.02): 0.05
The City Area (In km^2): 300
User Density (users/km^2): 200
Minimum SIR Required (In dB): 15
Please Enter The Sectorization method:
 a = omni-directional
 b = 120° sectorization
 c = 60° sectorization
 c

Cluster_size = 4
Total Number of Cells = 26
Traffic Intensity per Cell = 58.377 Erlang
Traffic Intensity per Sector = 9.7295 Erlang
Cell radius = 2.1074 Km
Ptx = 45.437 dBm
fx >>
```



Command Window

```
GOS (ex:0.02): 0.01
The City Area (In km^2): 90
User Density (users/km^2): 500
Minimum SIR Required (In dB): 16
Please Enter The Sectorization method:
 a = omni-directional
 b = 120° sectorization
 c = 60° sectorization
 b

Cluster_size = 7
Total Number of Cells = 43
Traffic Intensity per Cell = 26.6251 Erlang
Traffic Intensity per Sector = 8.875 Erlang
Cell radius = 0.89756 Km
Ptx = 31.9521 dBm
fx >>
```



Part B

1) Plot the cluster size versus SIRmin with range from 1dB to 30 dB.

Using this relation:

$$N \geq \frac{1}{3}\left(\sqrt[n]{i_o * 10^{\frac{SIR_{Ratio}}{10}}} + 1\right)^2$$

2) Plotting the number of calls and traffic intensity per cell versus grade of service (GoS) at user density 1400 users/km2 and SIRmin = 19 dB.

Using this relation

$$\text{Total number of Channels per sector} = \left\lfloor \frac{S}{N * \text{number of sectors}} \right\rfloor$$

$$P_r = \frac{\frac{A^c}{C!}}{\sum_{K=0}^{c} \frac{A^k}{K!}} = GOS$$

$$A\ \text{per cell} = A\ \text{per sector} * \text{number of sectors}$$

$$\text{Total Number of cells} = \left\lceil \frac{A}{A\ \text{Per Cell}} \right\rceil$$

3) Plotting the number of calls and traffic intensity per cell versus grade of service (GoS) at user density 1400 users/km2 and SIRmin = 14 dB.

Using this relation

$$\text{Total number of Channels per sector} = \left\lfloor \frac{S}{N * \text{number of sectors}} \right\rfloor$$

$$P_r = \frac{\frac{A^c}{C!}}{\sum_{K=0}^{c} \frac{A^k}{K!}} = GOS$$

$$A\ \text{per cell} = A\ \text{per sector} * \text{number of sectors}$$

$$\text{Total Number of cells} = \left\lceil \frac{A}{A\ \text{Per Cell}} \right\rceil$$

4) Plotting the number of cells and cell radius versus user density at SIRmin = 14 dB and grade of service (GoS) = 2%.

$$\text{Total } number\ of\ Channels\ per\ sector = \left\lfloor \frac{S}{N * number\ of\ sectors} \right\rfloor$$

$$P_r = \frac{\frac{A^C}{C!}}{\sum_{K=0}^{c} \frac{A^k}{K!}} = GOS$$

$$Total\ Number\ of\ cells\ = \left\lceil \frac{A}{A\ Per\ Cell} \right\rceil$$

$$Cell\ Radius\ (R) = \sqrt{\frac{Cell\ Area}{\frac{3\sqrt{3}}{2}}}$$

5) Plotting the number of cells and cell radius versus user density at SIRmin = 19 dB and grade of service (GoS) = 2%.

$$\text{Total } number\ of\ Channels\ per\ sector = \left\lfloor \frac{S}{N * number\ of\ sectors} \right\rfloor$$

$$P_r = \frac{\frac{A^C}{C!}}{\sum_{K=0}^{c} \frac{A^k}{K!}} = GOS$$

$$Total\ Number\ of\ cells\ = \left\lceil \frac{A}{A\ Per\ Cell} \right\rceil$$

$$Cell\ Radius\ (R) = \sqrt{\frac{Cell\ Area}{\frac{3\sqrt{3}}{2}}}$$

6) MATLAB code for part 2:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%Part 2%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(1)Cluster Size Vs SIR%
% Range of SIRmin_dB values to plot

SIRmin_dB_range = 1:0.001:30;
path_loss_exponent = 4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialize arrays to store cluster sizes for each sectorization method
cluster_sizes_a = zeros(size(SIRmin_dB_range)); %
cluster_sizes_b = zeros(size(SIRmin_dB_range));
cluster_sizes_c = zeros(size(SIRmin_dB_range));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:length(SIRmin_dB_range)
    SIRmin_dB = SIRmin_dB_range(i);

    % Calculate SIR ratio in dB
    SIR_ratio = 10^(SIRmin_dB/10);

    % Calculate the cluster size for omni-directional sectorization
    i0 = 6;
    cluster_sizes_a(i) = calculate_cluster_size(i0, SIR_ratio, path_loss_exponent);
    % Calculate the cluster size for 120° sectorization
    i0 = 2 ;
    cluster_sizes_b(i) = calculate_cluster_size(i0, SIR_ratio, path_loss_exponent);
    % Calculate the cluster size for 60° sectorization
    i0 = 1;
    cluster_sizes_c(i) = calculate_cluster_size(i0, SIR_ratio, path_loss_exponent);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot the cluster size versus SIRmin_dB for all sectorization methods
figure;
plot(SIRmin_dB_range, cluster_sizes_a, 'LineWidth', 2);
hold on;
plot(SIRmin_dB_range, cluster_sizes_b, 'LineWidth', 2);
plot(SIRmin_dB_range, cluster_sizes_c,  'LineWidth', 2);
ylabel('Cluster Size');
xlabel('Minimum SIR Required (dB)');
title('Cluster Size vs. Minimum SIR Required for Different Sectorization Methods');
legend('Omni-directional', '120° sectorization', '60° sectorization', 'Location',
'Northwest');
grid on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%(2)&(3)Number of cells and Intensity Vs GOS%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

SIRmin_dB = input(['Number of cells and Intensity Vs GOS \n...' ...
    ' Minimum SIR Required(In dB): ']); % Use 14dB then use 19dB
%Constants
city_area = 100;%km^2
user_density= 1400;%users/km^2
traffic_per_user = 0.025; % Traffic intensity per user in Erlangs
path_loss_exponent = 4;
channels = 340;
% Range of GOS_values values to plot
GOS_values = 0.01:0.01:0.3;
SIR_ratio = 10^(SIRmin_dB/10);
```

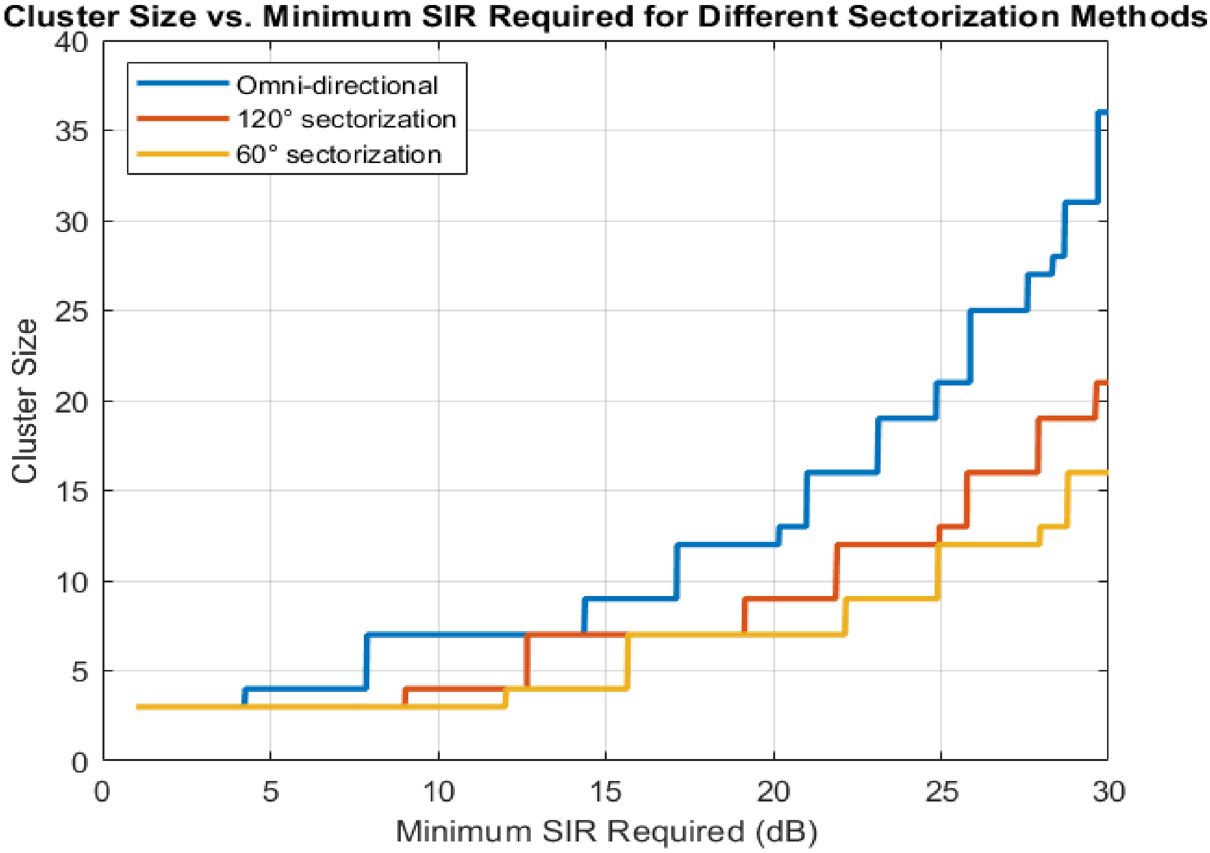
```
% Initialize arrays to store cluster sizes for each sectorization method
n_i0 = [6 2 1]; n_sectors = [1 3 6];
Num_cells = zeros(length(n_i0), length(GOS_values));
traffic_intensity_per_cell = zeros(length(n_i0), length(GOS_values));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for X = 1 : length(n_i0)
    i0 = n_i0(X); sectors = n_sectors(X);
    Cluster_size = calculate_cluster_size(i0, SIR_ratio, path_loss_exponent);
    num_channels_per_sector = floor(channels / (Cluster_size * sectors));
    for Y = 1:length(GOS_values)
        GOS = GOS_values(Y);

        num_channels_per_sector = floor(channels / (Cluster_size * sectors));
    %Solving The Erling B equation using fzero function
        fun = @(A) GOS - (A^num_channels_per_sector/factorial(num_channels_per_sector))
...
        / sum(A.^((0:num_channels_per_sector))./factorial(0:num_channels_per_sector));
        traffic_intensity_per_sector = fzero(fun, [0, 1000]);


        traffic_intensity_per_cell(X,Y) = traffic_intensity_per_sector * sectors;
        total_traffic_intensity = user_density * city_area * traffic_per_user;
        Num_cells(X,Y) = ceil((total_traffic_intensity) /
traffic_intensity_per_cell(X,Y));
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot the number of cells vs GOS
figure;
plot(GOS_values, Num_cells(1,:), 'r', GOS_values, Num_cells(2,:),...
    'g', GOS_values, Num_cells(3,:), 'b', 'LineWidth', 2);
title(sprintf('Number of Cells vs GOS (SIRmin = %d dB)', SIRmin_dB));
xlabel('GOS');
ylabel('Number of Cells');
legend('Omni directional', '120° sectorization', '60° sectorization');
grid on;
% Plot the traffic intensity per cell vs GOS
figure;
plot(GOS_values, traffic_intensity_per_cell(1,:), 'r', GOS_values,...
    traffic_intensity_per_cell(2,:), 'g', GOS_values, traffic_intensity_per_cell(3,:),
'b', 'LineWidth', 2);
title(sprintf('Traffic Intensity per Cell vs GOS (SIRmin = %d dB)', SIRmin_dB));
xlabel('GOS');
ylabel('Traffic Intensity per Cell');
legend('Omni directional', '120° sectorization', '60° sectorization');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%(4)&(5) Cell Redius and Number of cells Vs User Density%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SIRmin_dB = input(['Cell Redius and Number of cells Vs User Density \n ...' ...
    ' Minimum SIR Required(In dB): ']); % Use 14dB then use 19dB

%Constants%
city_area = 100;%km^2
GOS=0.02;
traffic_per_user = 0.025; % Traffic intensity per user in Erlangs
path_loss_exponent = 4;
```

```
channels = 340;
user_density = 100:2000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SIR_ratio = 10^(SIRmin_dB/10);
% Initialize arrays to store cluster sizes for each sectorization method%
n_i0 = [6 2 1]; n_sectors = [1 3 6];
Num_cells = zeros(length(n_i0), length(user_density));
cell_red = zeros(length(n_i0), length(user_density));
for X = 1 : length(n_i0)
    i0 = n_i0(X); sectors = n_sectors(X);
    Cluster_size = calculate_cluster_size(i0, SIR_ratio, path_loss_exponent);
    num_channels_per_sector = floor(channels / (Cluster_size * sectors));
%Solving The Erling B equation using fzero function
      fun = @(A) GOS - (A^num_channels_per_sector/factorial(num_channels_per_sector))
...
        / sum(A.^((0:num_channels_per_sector))./factorial(0:num_channels_per_sector));
        traffic_intensity_per_sector = fzero(fun, [0, 1000]);

        traffic_intensity_per_cell = traffic_intensity_per_sector * sectors;
        for Y = 1:length(user_density)
            total_traffic_intensity = user_density(Y) * city_area * traffic_per_user;
            Num_cells(X,Y) = ceil((total_traffic_intensity)) /
traffic_intensity_per_cell;
            cell_Area = city_area/Num_cells(X,Y);
            cell_red(X,Y) = sqrt((2*cell_Area)/(3*sqrt(3)));
        end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot the number of cells versus user density%
figure();
plot(user_density, Num_cells(1,:), 'b', 'LineWidth', 2);
hold on;
plot(user_density, Num_cells(2,:), 'r', 'LineWidth', 2);
plot(user_density, Num_cells(3,:), 'g', 'LineWidth', 2);
grid on;
xlabel('User density (users/km^2)', 'FontSize', 12);
ylabel('Number of cells', 'FontSize', 12);
legend('Omni directional', '120° sectorization', '60° sectorization');
title(['Number of cells vs. user density (SIR_{min} = ' num2str(SIRmin_dB) ' dB)'],
'FontSize', 14);
% Plot the cell radius versus user density
figure();
plot(user_density, cell_red(1,:), 'b', 'LineWidth', 2);
hold on;
plot(user_density, cell_red(2,:), 'r', 'LineWidth', 2);
plot(user_density, cell_red(3,:), 'g', 'LineWidth', 2);
grid on;
xlabel('User density (users/km^2)', 'FontSize', 12);
ylabel('Cell radius (km)', 'FontSize', 12);
legend('Omni directional', '120° sectorization', '60° sectorization');
title(['Cell radius vs. user density vs. user density ...' ...
    ' (SIR_{min} = ' num2str(SIRmin_dB) ' dB)'], 'FontSize', 14);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%Functions%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%Function used in part1%%%%%%%%%%%%%%%%%%%%%%%%%%
function L=Hata(f,hm,hb,distance)
CH =0.8+(1.1*log10(f)-0.7)*hm-1.5 *log10(f);
```

```
L=69.55 + 26.16 * log10(f) - 13.82 * log10(hb) - CH + (44.9 - 6.55 * log10(hb)) *
log10(distance);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%Function used in part1%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Cluster_size = calculate_cluster_size(i0, SIR_ratio, path_loss_exponent)
    Q = (SIR_ratio*i0)^(1/path_loss_exponent);
    A_TEMP = ((Q+1).^2)/3;
    Cluster_size = ceil(A_TEMP);
    found_n = false;  % flag to indicate if a solution has been found
    while ~found_n
        for j = 0:Cluster_size
            for k = 0:Cluster_size
                if j^2 + j*k + k^2 == Cluster_size
                    found_n = true;
                    break;
                end
            end
            if found_n
                break;
            end
        end
        if ~found_n
            Cluster_size = Cluster_size + 1;
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7) Graphs:

1. Plot the cluster size versus SIR_{min} with range from 1dB to 30 dB.



Note:

- As cluster size increases, the signal-to-interference ratio (SIR) also rises.
- As the cluster size is limited by this equation

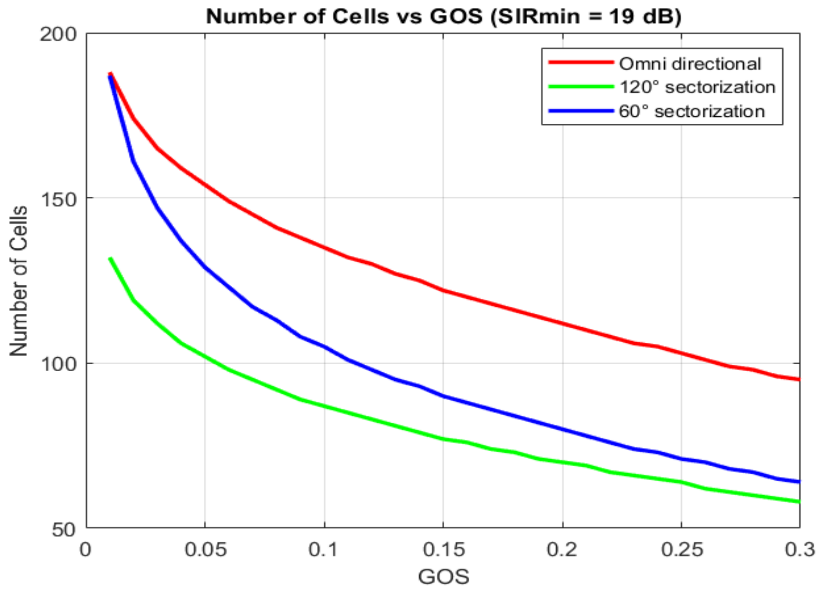
$$N = i^2 + ik + k^2$$

- So, it takes discrete values.
- when SIR increases i_o is directly proportional to N.
- Signal-to-interference ratio (SIR) and N are directly proportional.
- For the same SIR

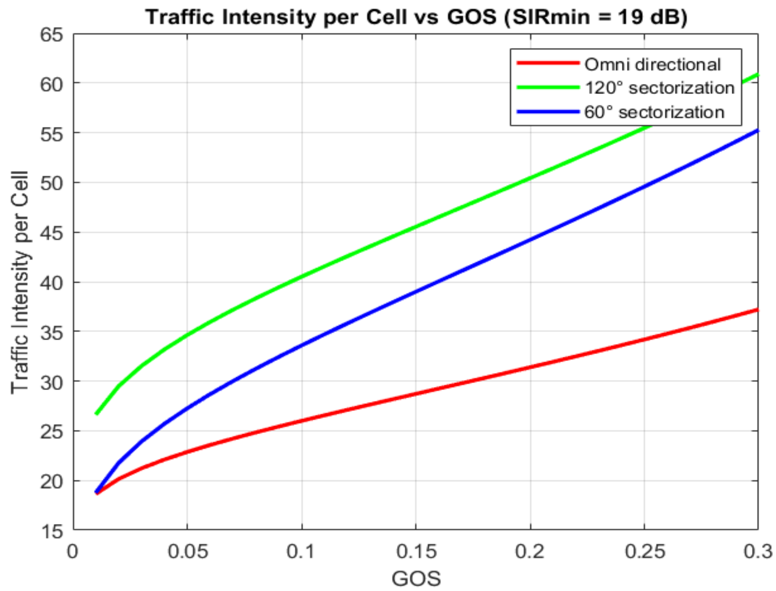
$$N_{Omni-directional} > N_{120^o Sectorization} > N_{60^o Sectorization}$$
$$C_{Omni-directional} < C_{120^o Sectorization} < C_{60^o Sectorization}$$

2. For SIR_{min} = 19dB & user density= 1400 users/km2

❖ *Plot the number of cells versus GOS (1% to 30%).*



❖ *Plot the traffic intensity per cell versus GOS (1%to 30%).*

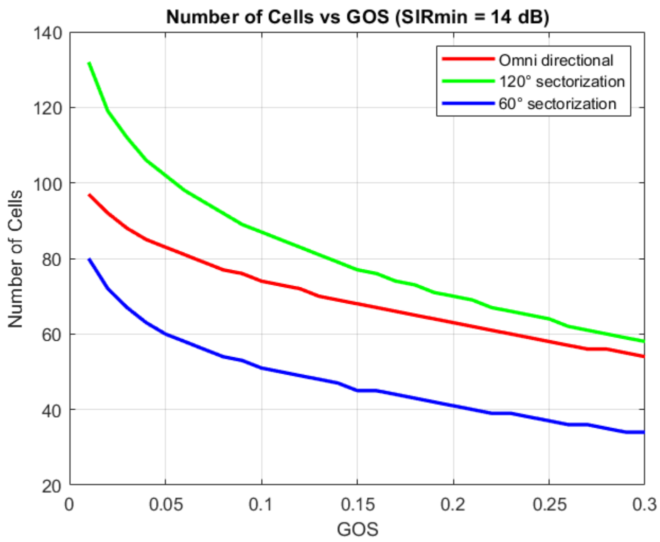


Note

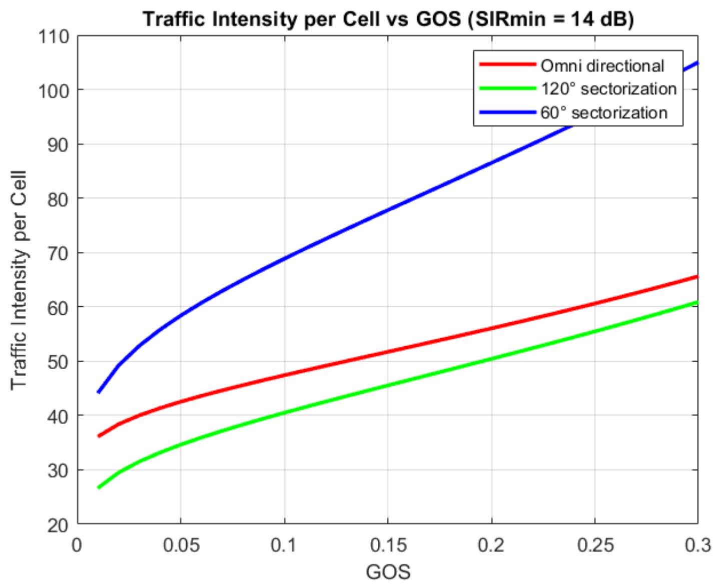
- With higher grade of service (GoS), the number of cells reduces roughly *exponentially*.
- With higher grades of service, there is more traffic per cell.
- *120 sectorization can offer better frequency reuse and higher capacity at the expense of complexity and potential interference compared to 60 sectorization. The plot shows that 120 sectorization is more efficient in terms of spectrum utilization and can reduce the number of cells needed, but the choice should be based on careful analysis of network requirements and trade-offs.*

3. At SIR_{min}= 14dB & user density= 1400 users/km2

- *Plot the number of cells versus GOS (1% to 30%).*



- *Plot the traffic intensity per cell versus GOS (1% to 30%).*

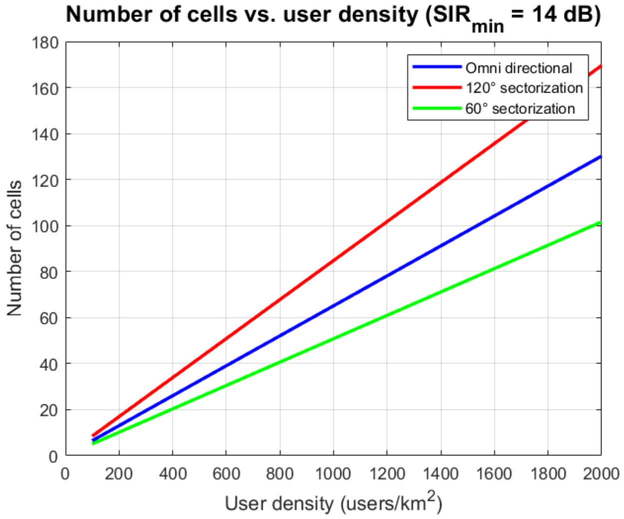


Note:

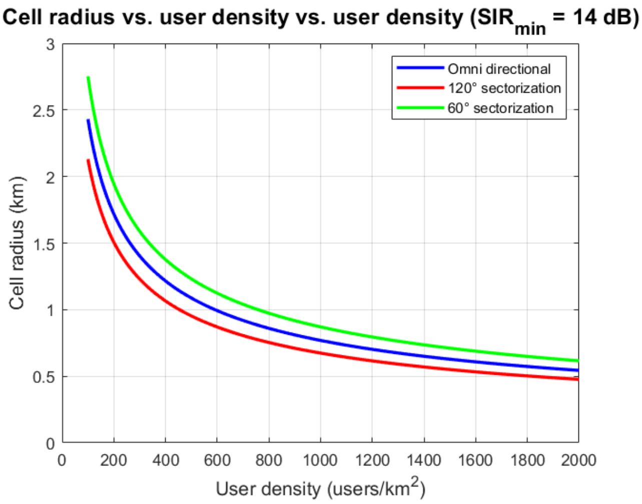
- *With higher service levels comes fewer cells, but higher service levels also mean more traffic in each cell.*
- *The number of cells is higher in the graphs at SNR = 19 dB than in the graphs at SNR = 14 dB, and the traffic intensity per cell is higher in the graphs at SNR = 14 dB than in the graphs at SNR = 19 dB.*
- *The 60-degree sectorization method has the fewest cells overall and the highest traffic density per cell of the three sectorization techniques.*
- *With decreasing the SIR* $60^{o}\,Sectorization\ is\ better\ than\ 120^{0}Sectorization\ for\ number\ of\ cells\ and\ traffic\ intensity\ per\ cell\ for\ the\ same\ GOS.$

4. At $\boldsymbol{SIR_{min}}$= 14dB & GOS= 2%,

- *Plot the number of cells versus user density (100 to 2000 users/km2).*



- *Plot the cell radius versus user density (100 to 2000 users/km2).*



Note:

- *With higher user density, the number of cells increases linearly as:*

$$Number\ of\ Cells\ = \lceil \frac{Total\ traffic\ intensity}{Traffic\ intensity\ per\ cell} \rceil = \lceil \frac{User\ Density\ *\ City\ Area\ *\ Au}{Traffic\ intensity\ per\ cell} \rceil$$

- *Cell radius decreases with increasing user density as*

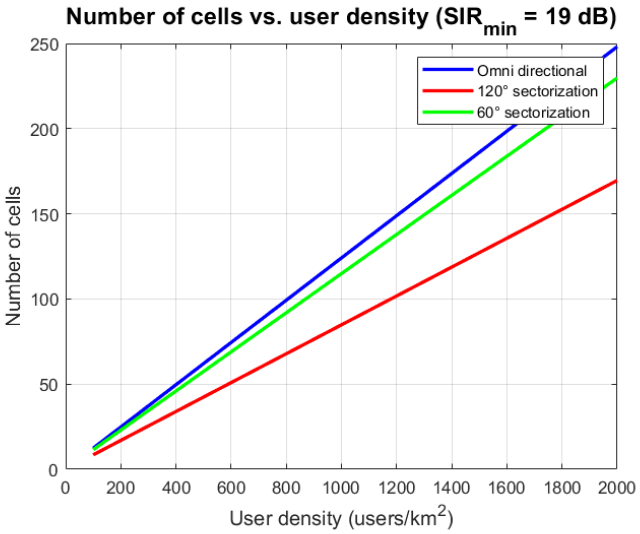
$$Cell\ Area\ = \frac{City\ Area}{Number\ of\ Cells}$$

$$Cell\ Radius\ (R) = \sqrt{\frac{Cell\ Area}{\frac{3\sqrt{3}}{2}}}$$

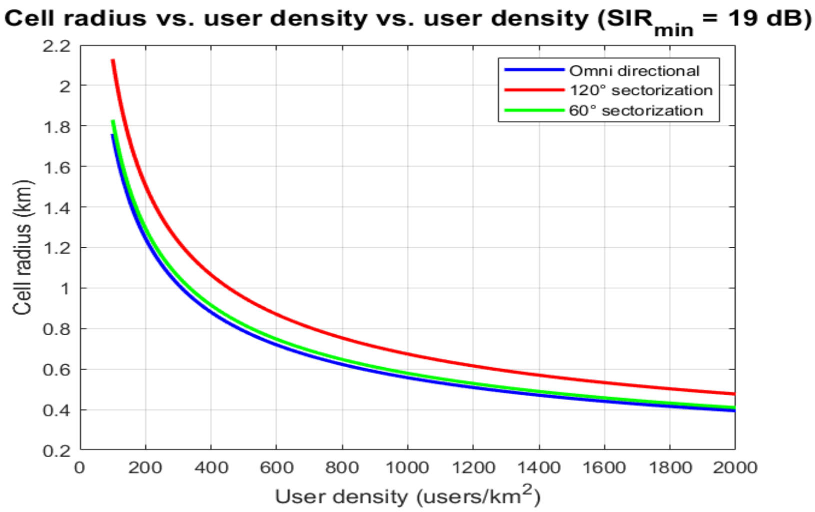
- *The cell radius therefore has an inverse relationship with the number of cells.*

5. At SIR_{min}= 19dB & GOS= 2%,

- *Plot the number of cells versus user density (100 to 2000 users/km2).*



- *Plot the cell radius versus user density (100 to 2000 users/km2).*



Note:

- *In the case of 19 dB, the number of cells is greater than in the case of 14 dB, and the cell radius is larger than in the case of 19 dB.*
- *With decreasing the SIR $60^o Sectorization\ is\ better\ than\ 120^0 Sectorization\ for\ number\ of\ cells\ and\ cell\ radius\ per\ cell\ for\ the\ same\ user\ density.$*