Ain Shams University
Faculty of Engineering
Electronics & Comm. Dept.

ECE 353s
3rd Year Electrical Engineering
Spring 2024

Project Report

Name	ID	Sec.
Abanoub Emad Hanna Kamel	2002075	4
Yousif Ahmed Ali Mohamed	2001902	4
Shady Emad Zaki	2001553	3
Omar Mahmoud ElSayed Gabr	2000476	2

Hand Analysis & Functions Explanation

Cluster Size

Since $SIR_{\min dB}$ & number of sectors are given, we can calculate N from the following equation:

$$SIR_{\min(ratio)} = \frac{1}{Sectorization} * \left(\sqrt{3\,N} - 1\right)^{n}$$

Note: interference variable is determined from sectorization method to follow the stated equation.

```
function N = cluster_size(S, interference, SIR_dB, n)
    SIR = 10^(SIR_dB/10);
    N = ceil((1/3)*((interference*SIR)^(1/n) + 1)^2);
    N = valid_N(N);
end
```

Then we have to check for nearest valid cluster size from the following equation:

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

```
function N = valid_N(n)
    limit = 100; % limit of i & k
    i = 0:limit;
    valid = [];
    for k = 0:limit
        valid = [valid (i.^2 + k^2 + i.*k)];
    end
    valid = unique(sort(valid));
    while ~(ismember(n,valid))
        n = n + 1;
        if n > 30000
            print ('N is too large');
            break;
        end
    end
    N = n;
end
```

Traffic Intensity

Number of channels per sector can be calculated from total number of channels (S) by:

$$K = \frac{S}{N * Sectors}$$

Instead of using Erlang charts, we can use probability of blocking equation:

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

```
function A = inverlangb(c, gos)
    fun = @(A) gos - (A^c/factorial(c)) / sum(A.^((0:c))./factorial(0:c));
    A = fzero(fun, [0, 1000]);
end
```

The previous function returns the traffic intensity per sector, then we can calculate traffic intensity per cell by:

$$A_{cell} = A_{sector} * number\ of\ sectors$$

```
function [Acell, Asector] = traffic_intensity(S, N, sectors, GOS)
    K = floor(S / (N * sectors));
    Asector = inverlangb(K, GOS);
    Acell = Asector * sectors;
end
```

Note: We have to floor so that no extra bandwidth is taken.

Number of Cells

Users per cell can be calculated by:

$$A_{cell} = A_u\ U_{cell}$$

Total Number of users can be calculated by:

$$Total\ Users = User\ denisty\ *\ Area\ of\ city$$

Total number of cells can be calculated by:

$$Total\ Number\ of\ Cells = \frac{Total\ Users}{Users/Cell}$$

```
function Cells = no_of_cells(Area, User_Den, Acell, Au)
    Users_Per_cell = Acell / Au; % number of users per cell
    Total_Users = User_Den * Area; % total number of users
    Cells = ceil(Total_Users / Users_Per_cell); % total number of cells
end
```

Note: We have to ceil so that there are no dead zones.

Cell radius

From the given city area, we can calculate the radius by:

$$Area\ of\ cell = \frac{city\ area}{number\ of\ cells}$$

$$\text{Area of cell} = \frac{3}{2}\sqrt{3}\ R^2 \ \rightarrow \text{R} = \sqrt{\frac{Area\ of\ cell}{\frac{3}{2}\sqrt{3}}}$$

```
function R = radius(Area, Cells)
    Area_Per_cell = Area / Cells;
    R = sqrt(Area_Per_cell / (3 * sqrt(3) / 2));
end
```

Hata Function

The Hata model predicts the path loss in different environments. In our case urban-medium city is assumed.

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

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

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:

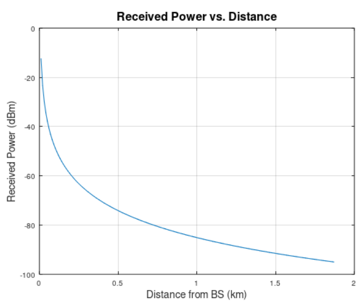
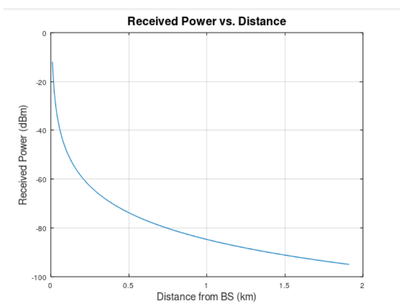
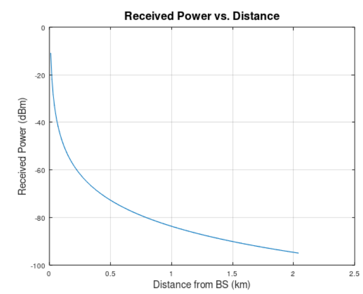
$$P_{Rx} = P_{Tx} - \ Path\ loss$$

Reference: 234679133.pdf (core.ac.uk)

```
function L = Hata(f,hm,hb,d)
    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(d);
end
```

Received Power vs Distance

- GOS = 0.05 (5%)
- City Area = 1000 km^2
- User Density = 100 $user/km^2$
- SIRmin_dB= 20 dB

<table>
<tr><th>Omni-directional</th><th>120 Sectorization</th><th>60 Sectorization</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>What is the value of GOS ?
0.05
What is the city Area ?
1000
What is the user denisty ?
100
What is the minimum SIR_dB ?
20
Choose sectorization method (a-b-c).
a) omnidirectional
b) 120 sectorization
c) 60 sectorization
a
Cluster size = 12
Number of cells = 110
Cell radius = 1.87059 km
Traffic intensity per cell = 22.8672 Erlang
Traffic intensity per sector = 22.8672 Erlang
Base station transmitted power = 43.5537 dBm
Choose sectorization method (a-b-c).
a) omnidirectional
b) 120 sectorization
c) 60 sectorization</td>
<td>What is the value of GOS ?
0.05
What is the city Area ?
1000
What is the user denisty ?
100
What is the minimum SIR_dB ?
20
Choose sectorization method (a-b-c).
a) omnidirectional
b) 120 sectorization
c) 60 sectorization
b
Cluster size = 9
Number of cells = 105
Cell radius = 1.91461 km
Traffic intensity per cell = 23.8502 Erlang
Traffic intensity per sector = 7.95006 Erlang
Base station transmitted power = 43.9212 dBm</td>
<td>What is the value of GOS ?
0.05
What is the city Area ?
1000
What is the user denisty ?
100
What is the minimum SIR_dB ?
20
Choose sectorization method (a-b-c).
a) omnidirectional
b) 120 sectorization
c) 60 sectorization
c
Cluster size = 7
Number of cells = 92
Cell radius = 2.04541 km
Traffic intensity per cell = 27.2578 Erlang
Traffic intensity per sector = 4.54296 Erlang
Base station transmitted power = 44.9653 dBm
Incorrect input (options : a || b || c)
Choose sectorization method (a-b-c).
a) omnidirectional
b) 120 sectorization
c) 60 sectorization</td>
</tr>
</table>

Comments:

- Received power decreases as distance increases, which is predicted by this formula:

$$P_{Rx} = P_{Tx} G_{Tx} G_{Rx} \left(\frac{H_{Tx} * H_{Rx}}{d^2} \right)^2$$

- Sectorization type does not affect the above graph since transmitted power is kept constant.

Complete Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Part A %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%% Inverse Erlang function %%%%%%%%%%%%%%%%%%%%%%%%%%

function A = inverlangb(c, gos)
    fun = @(A) gos - (A^c/factorial(c)) / sum(A.^((0:c))./factorial(0:c));
    A = fzero(fun, [0, 1000]);
end

%%%%%%%%%%%%%%%%%%%%%%%%% Hata function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function L = Hata(f,hm,hb,d)
    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(d);
end

%%%%%%%%%%%%%%%%%%%%%%% Valid N function %%%%%%%%%%%%%%%%%%%%%%%%%%%%

function N = valid_N(n)
    limit = 100; % limit of i & k
    i = 0:limit;
    valid = [];
    for k = 0:limit
        valid = [valid (i.^2 + k^2 + i.*k)];
    end
    valid = unique(sort(valid));
    while ~(ismember(n,valid))
        n = n + 1;
        if n > 30000
            print ('N is too large');
            break;
        end
    end
    N = n;
end

%%%%%%%%%%%%%%%%%%%%% Cluster Size function %%%%%%%%%%%%%%%%%%%%%%%%%%

function N = cluster_size(S, interference, SIR_dB, n)
    SIR = 10^(SIR_dB/10);
    N = ceil((1/3)*((interference*SIR)^(1/n) + 1)^2);
    N = valid_N(N);
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%% Acell function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Acell, Asector] = traffic_intensity(S, N, sectors, GOS)
    K = floor(S / (N * sectors));
    Asector = inverlangb(K, GOS);
    Acell = Asector * sectors;
end

%%%%%%%%%%%%%%%%%%%%%%% No. of Cells function %%%%%%%%%%%%%%%%%%%%%%%

function Cells = no_of_cells(Area, User_Den, Acell, Au)
    Users_Per_cell = Acell / Au; % number of users per cell
    Total_Users = User_Den * Area; % total number of users
    Cells = ceil(Total_Users / Users_Per_cell); % total number of cells
end

%%%%%%%%%%%%%%%%%%%%%%%%% Radius function %%%%%%%%%%%%%%%%%%%%%%%%%%%%

function R = radius(Area, Cells)
    Area_Per_cell = Area / Cells;
    R = sqrt(Area_Per_cell / (3 * sqrt(3) / 2));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialization %%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear;
clc;
pkg load communications;

S = 340;  % Total number of channels : S= N*K
freq = 900; % Frequency in MHz
sensitivity = -95; % in db
Au = 0.025; % in erlangs
n = 4; % path loss exponent
h_BS = 20; % Base Station height
h_MS = 1.5; % Mobile Station height

%%%%%%%%%%%%%%%%%%%%%%%% Asking for inputs %%%%%%%%%%%%%%%%%%%%%%%%%%%%

prompt = "What is the value of GOS ? \n";
GOS = input(prompt);
```

```
prompt= "What is the city Area ? \n";
Area = input(prompt);

prompt = "What is the user denisty ? \n";
User_Den = input(prompt);

prompt = "What is the minimum SIR_dB ? \n";
SIR_dB = input(prompt);

prompt = "Choose sectorization method (a-b-c).\na) omnidirectional\nb) 120
sectorization\nc) 60 sectorization \n" ;
sectorization = input(prompt, 's');
sf = 0; % flag for sectorization method check

while ~sf
    switch sectorization
        case 'a'
            sectors = 1;
            interference = 6;
            sf = 1;
        case 'b'
            sectors = 3;
            interference = 2;
            sf = 1;
        case 'c'
            sectors = 6;
            interference = 1;
            sf = 1;
        otherwise
            fprintf ('Incorrect input (options : a || b || c) \n');
            sectorization = input(prompt, 's');
    end
end

%%%5%%%%%%%%%%%%%%%%%%%%%%%%% Calculations %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = cluster_size(S, interference, SIR_dB, n); % cluster size
[Acell, Asector] = traffic_intensity(S, N, sectors, GOS); % traffic intensity per
cell
Cells = no_of_cells(Area, User_Den, Acell, Au); % number of cells
Cell_Radius = radius(Area, Cells); % cell radius

path_loss = Hata(freq, h_MS, h_BS, Cell_Radius);
P_tx = path_loss + sensitivity;
```

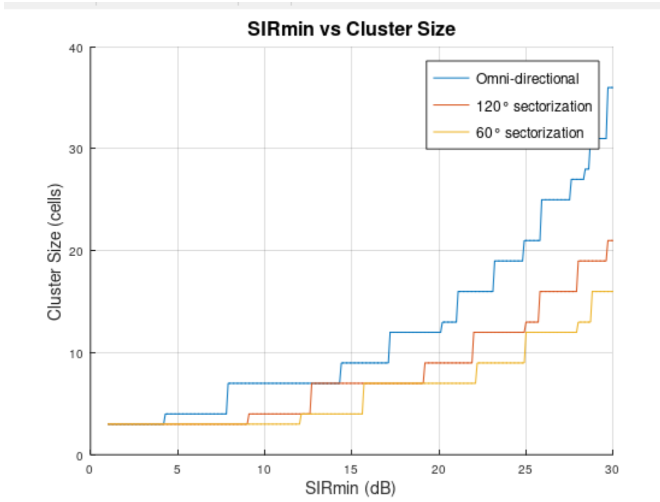
```
fprintf ('Cluster size = %d \n', N);
fprintf ('Number of cells = %d \n', Cells);
fprintf ('Cell radius = %d km\n', Cell_Radius);
fprintf ('Traffic intensity per cell = %d Erlang\n', Acell);
fprintf ('Traffic intensity per sector = %d Erlang\n', Asector);
fprintf ('Base station transmitted power = %d dBm\n', P_tx);

%%%%%%%%%%%%%%%%%%%%%%%%% Plot Prx vs distance %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

D = 0:0.01:Cell_Radius;
L = Hata(freq, h_MS, h_BS, D);
Prx = P_tx - L;
figure(1);
plot(D, Prx);
xlabel('Distance from BS (km)', 'FontSize', 14);
ylabel('Received Power (dBm)', 'FontSize', 14);
title('Received Power vs. Distance', 'FontSize', 16);
grid on
```

Part B

$SIRmin$ Sweep



Comments

- We can notice that the graph is stair-shaped because N has discrete values given by this equation:

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

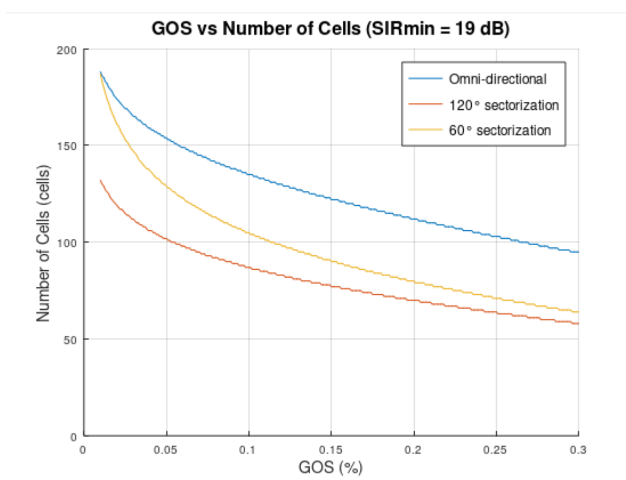
- Cluster size increases as the SIRmin increases as $N \; \alpha \; SIR_{min}$.

$$SIR_{\min(ratio)} = \frac{1}{Sectorization} * \left(\sqrt{3\,N} - 1\right)^n$$

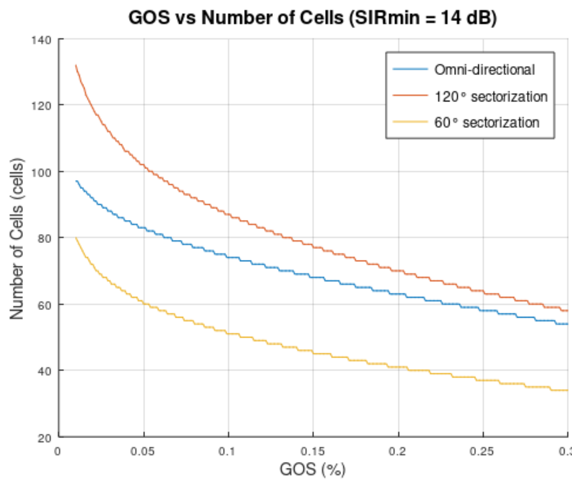
- For the same SIRmin, we can see that $N_{omni\text{-}directional} > N_{120} > N_{60}$, because propagation in more directions causes higher cochannel interference.

Number of cells vs GOS (User Density = 1400 users/km^2)

SIRmin = 19dB



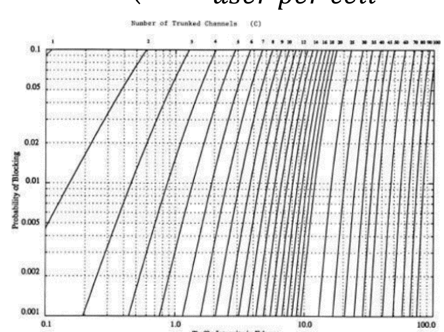
SIRmin = 14dB



Comments

- With higher grade of service (GOS), the number of cells reduces roughly exponentially.
- For any sectorization type:
  - We have constant SIR_{min} → constant cluster size (N)
  - As GOS increase for constant number of channels, Traffic intensity per cell (A_{cell}) increase (from Erlang graph)
  - As traffic intensity increase, User per cell increases ($User\ per\ cell = \frac{A_{cell}}{A_u\ (given\ constant)}$)
  - As user per cell increase, number of cells decrease:

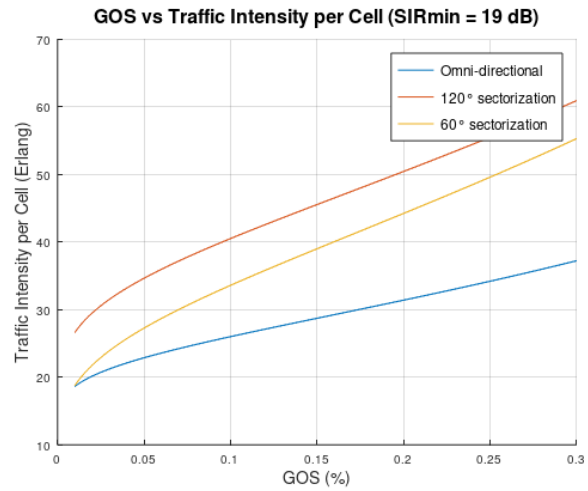
$$cells = ceil\left(\frac{Total\ number\ of\ users}{user\ per\ cell}\right)$$



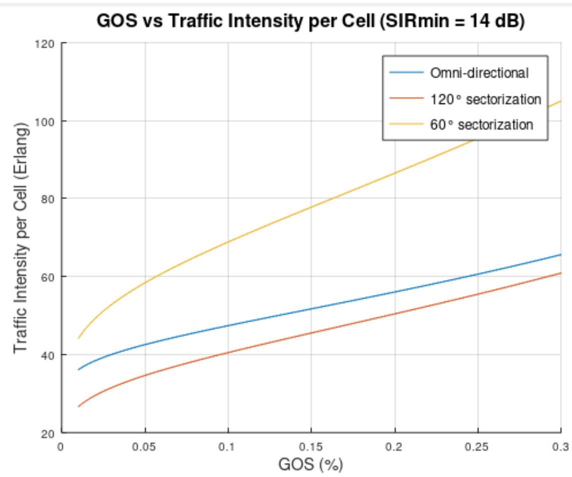
- The number of cells is higher in the graphs at SNR = 19 dB than in the graphs at SNR = 14 dB.

Traffic Intensity/Cell vs GOS (User Density = 1400 users/km^2)

SIRmin = 19dB



SIRmin = 14dB

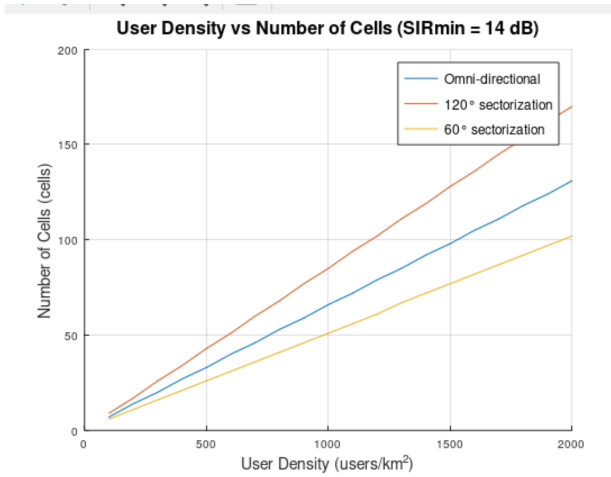


Comments

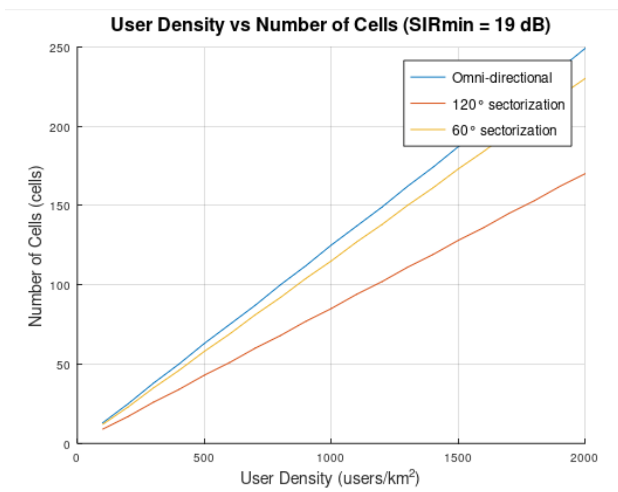
- Traffic intensity increases as GOS increases, as we can assign more users in one cell.
- The traffic intensity per cell is higher in the graphs at SIR = 14 dB than in the graphs at SIR = 19 dB, because the number of Trunked channels (k) for SIR =14 dB is larger than (k) for SIR = 19 dB, and traffic intensity per cell $\propto$ number of Trunked channels (k).

Number of cells versus user density (GOS = 2%)

SIRmin = 14dB



SIRmin = 19dB



Comments

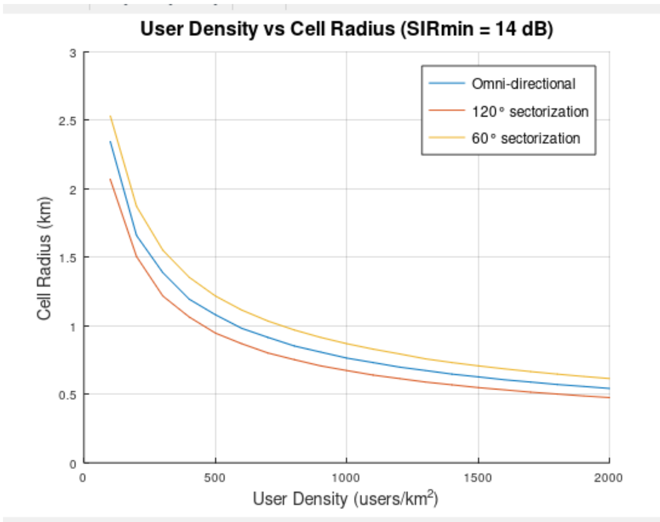
- With higher user density, the number of cells increases linearly as:

$$Number\ of\ Cells\ = \left[\frac{Total\ traffic\ intensity}{Traffic\ intensity\ per\ cell}\right] = \left[\frac{User\ Density\ *\ City\ Area\ *\ Au}{Traffic\ intensity\ per\ cell}\right]$$

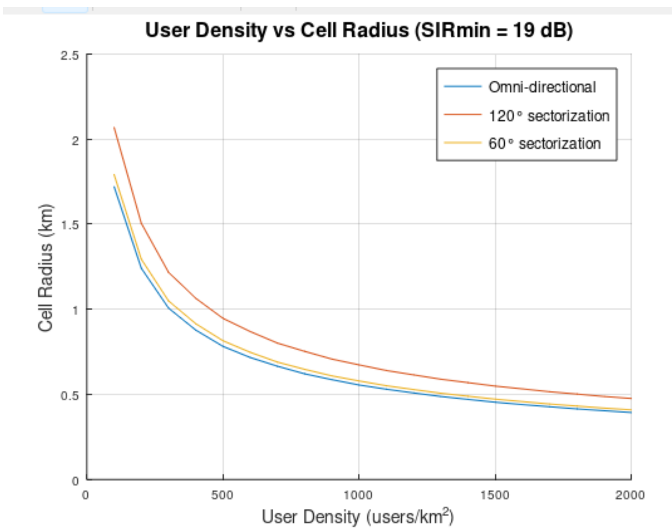
- The number of cells increases with higher SIR (Number of cells is greater with SIR = 19 dB).

Cell Radius vs User Fensity (GOS = 2%)

SIRmin = 14dB



SIRmin = 19dB



Comments

- Cell radius decreases with increasing user density as

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

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

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

- The cell radius increases with SIR (Cell radius is larger with SIR = 19 dB)

Complete Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Part B %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s_sweep = [1, 3, 6];    % sectorization sweep
i_sweep = [6, 2, 1];    % interference sweep

%%% 1) SIR Sweep

sir_sweep = 1:0.1:30;
N_sweep = [];
figure(2);
for j=1:3
    for i=1:length(sir_sweep)
        N_sweep(i) = cluster_size(S, i_sweep(j), sir_sweep(i), n);
    end
    hold on;
    plot(sir_sweep, N_sweep);
    xlabel('SIRmin (dB)', 'FontSize', 14);
    ylabel('Cluster Size (cells)', 'FontSize', 14);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization',
'FontSize', 12);
    end
    title('SIRmin vs Cluster Size', 'FontSize', 16);
    grid on;
end

%%%  2) SIRmin = 19dB & user density = 1400 users/km2


% Plot the number of cells & traffic intensity per cell versus GOS (1% to 30%).

gos_sweep = 1:0.1:30;
gos_sweep = gos_sweep / 100;
cells_sweep = [];
Acell_sweep = [];
Asector_sweep = [];
User_Den = 1400;

for j=1:3
    N = cluster_size(S, i_sweep(j), 19, n);
```

```
    for i=1:length(gos_sweep)
        [Acell_sweep(i), Asector_sweep(i)] = traffic_intensity(S, N, s_sweep(j), 
gos_sweep(i));
        cells_sweep(i) = no_of_cells(Area, User_Den, Acell_sweep(i), Au);
    end

    figure(3);
    hold on;
    plot(gos_sweep, cells_sweep);
    xlabel('GOS (%)', 'FontSize', 14);
    ylabel('Number of Cells (cells)', 'FontSize', 14);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization', 
'FontSize', 12);
    end
    title('GOS vs Number of Cells (SIRmin = 19 dB)', 'FontSize', 16);
    grid on;

    figure(4);
    hold on;
    plot(gos_sweep, Acell_sweep);
    xlabel('GOS (%)', 'FontSize', 14);
    ylabel('Traffic Intensity per Cell (Erlang)', 'FontSize', 14);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization', 
'FontSize', 12);
    end
    title('GOS vs Traffic Intensity per Cell (SIRmin = 19 dB)', 'FontSize', 16);
    grid on;

end


% 3) At SIRmin = 14dB & user density= 1400 users/km2

gos_sweep = 1:0.1:30;
gos_sweep = gos_sweep / 100;
cells_sweep = [];
Acell_sweep = [];
Asector_sweep = [];
User_Den = 1400;

for j=1:3
    N = cluster_size(S, i_sweep(j), 14, n);
    for i=1:length(gos_sweep)
```

```
        [Acell_sweep(i), Asector_sweep(i)] = traffic_intensity(S, N, s_sweep(j),
gos_sweep(i));
        cells_sweep(i) = no_of_cells(Area, User_Den, Acell_sweep(i), Au);
    end

    figure(5);
    hold on;
    plot(gos_sweep, cells_sweep);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization',
'FontSize', 12);
    end
    xlabel('GOS (%)', 'FontSize', 14);
    ylabel('Number of Cells (cells)', 'FontSize', 14);
    title('GOS vs Number of Cells (SIRmin = 14 dB)', 'FontSize', 16);
    grid on;

    figure(6);
    hold on;
    plot(gos_sweep, Acell_sweep);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization',
'FontSize', 12);
    end
    xlabel('GOS (%)', 'FontSize', 14);
    ylabel('Traffic Intensity per Cell (Erlang)', 'FontSize', 14);
    title('GOS vs Traffic Intensity per Cell (SIRmin = 14 dB)', 'FontSize', 16);
    grid on;
end

% 4) At SIRmin = 14dB & GOS= 2%

User_Den_sweep = 100:100:2000;
GOS = 2/100;
cells_sweep = [];
Acell_sweep = [];
Asector_sweep = [];
radius_sweep = [];


for j=1:3
    N = cluster_size(S, i_sweep(j), 14, n);
    for i=1:length(User_Den_sweep)
        [Acell_sweep(i), Asector_sweep(i)] = traffic_intensity(S, N, s_sweep(j),
GOS);
```

```
        cells_sweep(i) = no_of_cells(Area, User_Den_sweep(i), Acell_sweep(i),
Au);
        radius_sweep(i) = radius(Area, cells_sweep(i));
    end

    figure(7);
    hold on;
    plot(User_Den_sweep, cells_sweep);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization',
'FontSize', 12);
    end
    xlabel('User Density (users/km^2)', 'FontSize', 14);
    ylabel('Number of Cells (cells)', 'FontSize', 14);
    title('User Density vs Number of Cells (SIRmin = 14 dB)', 'FontSize', 16);
    grid on;

    figure(8);
    hold on;
    plot(User_Den_sweep, radius_sweep);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization',
'FontSize', 12);
    end
    xlabel('User Density (users/km^2)', 'FontSize', 14);
    ylabel('Cell Radius (km)', 'FontSize', 14);
    title('User Density vs Cell Radius (SIRmin = 14 dB)', 'FontSize', 16);
    grid on;
end

% 5) At SIRmin = 19dB & GOS= 2%

User_Den_sweep = 100:100:2000;
GOS = 2/100;
cells_sweep = [];
Acell_sweep = [];
Asector_sweep = [];
radius_sweep = [];


for j=1:3
    N = cluster_size(S, i_sweep(j), 19, n);
    for i=1:length(User_Den_sweep)
        [Acell_sweep(i), Asector_sweep(i)] = traffic_intensity(S, N, s_sweep(j),
GOS);
```

```
        cells_sweep(i) = no_of_cells(Area, User_Den_sweep(i), Acell_sweep(i), 
Au);
        radius_sweep(i) = radius(Area, cells_sweep(i));
    end

    figure(9);
    hold on;
    plot(User_Den_sweep, cells_sweep);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization', 
'FontSize', 12);
    end
    xlabel('User Density (users/km^2)', 'FontSize', 14);
    ylabel('Number of Cells (cells)', 'FontSize', 14);
    title('User Density vs Number of Cells (SIRmin = 19 dB)', 'FontSize', 16);
    grid on;

    figure(10);
    hold on;
    plot(User_Den_sweep, radius_sweep);
    if j == 3
        legend('Omni-directional', '120° sectorization', '60° sectorization', 
'FontSize', 12);
    end
    xlabel('User Density (users/km^2)', 'FontSize', 14);
    ylabel('Cell Radius (km)', 'FontSize', 14);
    title('User Density vs Cell Radius (SIRmin = 19 dB)', 'FontSize', 16);
    grid on;
end
```