A Simulator for a Network Engineer for Cell Organization.

I'm working on Mobile And Wireless Communication Lab Assignments in Python. I created a User Interface using **Python's Tkinter GUI**. A user should be able to provide the following input factors maintaining the constraints and getting the desired outputs.

Question No: 1

Input:

Area size to cover

Cell type: Macrocell, Microcell

Radius of each cell

Frequency reuse factor, N = I2 + J2 + (I*J);

where I, J = 0, 1, 2,3...

Example 1 [HAAS00]. Assume a system of 32 cells with a cell radius of 1.6 km, a total of 32 cells, a total frequency bandwidth that supports 336 traffic channels, and a reuse factor of $N = 7$. If there are 32 total cells, what geographic area is covered, how many channels are there per cell, and what is the total number of concurrent calls that can be handled? Repeat for a cell radius of 0.8 km and 128 cells.

Figure 4a shows an approximately square pattern. The area of a hexagon of radius R is $1.5R^2\sqrt{3}$. A hexagon of radius 1.6 km has an area of 6.65 km^2, and the total area covered is $6.65 \times 32 = 213$ km^2. For $N = 7$, the number of channels per cell is $336/7 = 48$, for a total channel capacity of $48 \times 32 = 1536$ channels. For the layout of Figure 4b, the area covered is $1.66 \times 128 = 213$ km^2. The number of channels per cell is $336/7 = 48$, for a total channel capacity of $48 \times 128 = 6144$ channels.

Output:

Number of cells required

Number of channels per cell

Total channel capacity

Total number of possible concurrent call.

In Coding Section, How can I execute Question-1?

At First, I am installing Python libraries.

```
from tkinter import *
from tkinter.ttk import Combobox
import math

window = Tk()
```

I have declared a class MyWindow:

```
    class MyWindow:
            def __init__(self, window):
```

#Set_Window & Window_Color

Here we have taken a window call MyWindow and declared the window size, and setting the background color of the window.

#TextLabel

In Python's GUI, I have used Text Label to display text. I have also changed the text's font and text size.

#TextBox / Field Entry

With the text boxes, I've taken field entry/input.

#Button Declaration

```
        self.btn1 = Button(window, text='Number Of Cells')
        self.btn2 = Button(window, text='Channels Per Cell')
        self.btn3 = Button(window, text='Total Channel Capacity')
        self.btn4 = Button(window, text='Concurrent cell')
```

#Label Position

In this section, I am positioning the Text Label in any position in the Python GUI window along the X and Y-axis, as well as positioning the input boxes via text box. Aside from that, I'm declaring the button colors and font sizes.

#Button Position

In this portion, I've specified where the buttons will appear in the Python GUI window, along the X and Y axes.

#Button Action:

```
def NumberOfCells(self):
self.t5.delete(0, 'end')                    #initially 0
        Total_area = float(self.t1.get())   #input Total_area as float
        radius = float(self.t2.get())       #input radius as float
        A1 = (2.6 * radius * radius)        #Area1 equation
        result = (Total_area / A1)    #Total area divided by Area1
self.t5.insert(END, int(result))    #click button show result as interger
def ChannelPerCell(self, event):
self.t5.delete(0, 'end')                        #initially 0
Total_Channel = int(self.t4.get())       #input TotalChanne as integer
Frequency_Reuse_Factor = int(self.t3.get())  #input FRF as interger
result = (Total_Channel / Frequency_Reuse_Factor)
self.t5.insert(END, int(result))    #click button show result as interger

def TotalChannelCapacity(self, event):
Total_area = float(self.t1.get())           #input Total_area as float
radius = float(self.t2.get())               #input radius as float
A1 = (2.6*radius*radius)                    #Area1 equation
result = (Total_area / A1)                  #Total area divided by Area1
Total_Channel = int(self.t4.get())       #input TotalChanne as integer
Frequency_Reuse_Factor = int(self.t3.get()) #input FRF as interger
result1 = (Total_Channel / Frequency_Reuse_Factor)
channel_capacity = (result*result1) #TotalArea*TotalChannel / FRF
self.t5.insert(END, str(channel_capacity))    #Result as string

def ConcurrentCell(self, event):
Total_area = float(self.t1.get())           #input Total_area as float
radius = float(self.t2.get())               #input radius as float
A1 = (2.6*radius*radius)                    #Area1 equation
result = (Total_area / A1)                  #Total area divided by Area1
Total_Channel = int(self.t4.get())       #input TotalChanne as integer
Frequency_Reuse_Factor = int(self.t3.get()) #input FRF as interger
result1 = (Total_Channel / Frequency_Reuse_Factor)
Total_calls=math.floor(result*result1)#TotalArea*TotalChannel / FRF
self.t5.insert(END, str(toatl_calls))       #Result as string
```

Question No: 2

Use the Okumara/Hata model to predict the path loss.

Input:

fc = carrier frequency in MHz from 150 to 1500 MHz
ht = height of transmitting antenna (base) from 30 to 300 m
hr = height of receiving antenna (mobile unit) from 1 to 10 m
d = propagation distance between antennas from 1 to 20 km
City size = Small/Medium, Large
Area type: Urban/Suburban, Open area

Example 2 [FREE97]. Let $f_c = 900$ MHz, $h_t = 40$ m, $h_r = 5$ m, and $d = 10$ km. Estimate the path loss for a medium-size city.

$$\begin{aligned} A(h_r) &= (1.1 \log 900 - 0.7)5 - (1.56 \log 900 - 0.8) \text{ dB} \\ &= 12.75 - 3.8 = 8.95 \text{ dB} \end{aligned}$$

$$\begin{aligned} L_{dB} &= 69.55 + 26.16 \log 900 - 13.82 \log 40 - 8.95 + (44.9 - 6.55 \log 40) \log 10 \\ &= 69.55 + 77.28 - 22.14 - 8.95 + 34.4 = 150.14 \text{ dB} \end{aligned}$$

Output: Predicted path loss in dB.

In Coding Section, How can I execute Question-2?

At First, I am installing Python libraries.

```
import math
from tkinter import *
from tkinter.ttk import Combobox

window = Tk()
```

I have declared a class MyWindow:

```
    class MyWindow:
            def __init__(self, window):
```

#Set_Window & Window_Color

Here we have taken a window call MyWindow and declared the window size, and setting the background color of the window.

#TextLabel
In Python's GUI, I have used Text Label to display text. I have also changed the text's font and text size.

#TextBox / Field Entry
With the text boxes, I've taken field entry/input.

#Button Declaration

```
self.b1 = Button(window, text='Path Loss(dB):',
    command=self.pathloss, fg='Blue', font=("Arial Bold", 14))
self.b1.place(x=250, y=410)
```

#Label Position
In this section, I am positioning the Text Label in any position in the Python GUI window along the X and Y-axis, as well as positioning the input boxes via text box. Aside from that, I'm declaring the button colors and font sizes.

#Function of Equation

```
    def pathloss(self):
      self.t7.delete(0, 'end')
      fc = float(self.t1.get())
      ht = float(self.t2.get())
      hr = float(self.t3.get())
      link_distance = float(self.t4.get())
      city = float(self.t5.get())
      area = float(self.t6.get())

      correction_factor = 0.0

      path_loss = 0.0

if city == 1:
          correction_factor = (0.8 + ((1.1 * math.log10(fc) - 0.7)
                              * hr) - (1.56 * math.log10(fc)))
```

```
    else:
        if (fc >= 150) and (fc <= 200):
            correction_factor = (
             8.29 * (math.pow(math.log10(1.54 * hr), 2))) - 1.1
        else:
            correction_factor = (
             3.2 * (math.pow(math.log10(11.75 * hr), 2))) - 4.97

path_loss = 69.55 + (26.16*math.log10(fc)) - (13.82*math.log10(ht))
        - \ correction_factor + \ ((44.9 - (6.55 * math.log10(ht)))
         * math.log10(link_distance))

    diff_loss = 0.0
    if area == 2:
        diff_loss = 2 * math.pow(math.log10((fc/28)), 2) + 5.4
    elif area == 3:
        diff_loss = 4.78 * \
            math.pow(math.log10(fc), 2) + 18.733 * \
            math.log10(fc) + 40.94

    path_loss -= diff_loss
  self.t7.insert(END, str(path_loss))
```

For a small/medium-sized city = 1, the correction factor is

$$A(h_r) = (1.1 \log f_c - 0.7)h_r - (1.56 \log f_c - 0.8)\ dB$$

And for a large city = 2, it is given by

$$A(h_r) = 8.29\ [\log(1.54\ h_r)]^2 - 1.1\ dB \quad \text{for } f_c \leq 300\ MHz$$

$$A(h_r) = 3.2\ [\log(11.75\ h_r)]^2 - 4.97\ dB \text{ for } f_c \geq 300\ MHz$$

To estimate the path loss in a suburban area, the formula for urban path loss in Equation (1) is modified as

$$L_{dB}(suburban) = L_{dB}(urban) - 2[\log(f_c / 28)]^2 - 5.4$$

And for the path loss in open areas, the formula is modified as

$$L_{dB}(open) = L_{dB}(urban) - 4.78(\log f_c)^2 - 18.733(\log f_c) - 40.98$$