Introduction to Wireless and Mobile Networking
Hw3 – Report

B03901026 許凱傑

I MATLAB code

1 Main Function

(1) Parameter setting

```
% Parameter setting
ISD     = 500;          % inter site distance
side    = ISD/sqrt(3);
num_MS  = 100;
sim_time = 900;
T       = 27 + 273.15;
B       = 10e6;
H_BS    = 1.5;          % height of Base station
H_B     = 50;           % height of building
H_R     = H_BS + H_B;
P_BS    = 33 - 30;      % BS power = 33 dBm
G_R_dB = 14;
G_R = fromdB(G_R_dB);

BS_X = side*[-3,-3,-3,-1.5,-1.5,-1.5,-1.5,0,0,0,0,0,1.5,1.5,1.5,1.5,3,3,3];
BS_Y = ISD*[-1,0,1,-1.5,-0.5,0.5,1.5,-2,-1,0,1,2,-1.5,-0.5,0.5,1.5,-1,0,1];
```

(2) get the border of the observed 19cell-map and the surrounding 6 19cell-map.

```
%% 3-1
figure;
[borderX, borderY] = cellmap(side,BS_X,BS_Y,0,0,1,1);
Xmax = max(borderX);
Ymax = max(borderY);
title('Figure B-1');
xlabel('Distance(m)'), ylabel('Distance(m)');
axis([-1.1*Xmax, 1.1*Xmax,-1.1*Ymax, 1.1*Ymax])
hold off;
offsetX = side*[4.5, 7.5, 3, -4.5, -7.5, -3];
offsetY = ISD*[3.5, -0.5, -4, -3.5, 0.5, 4];
for i = 1:6
   [outX{i} , outY{i}] = cellmap(side, BS_X, BS_Y, offsetX(i), offsetY(i), 0, 0);
end
```

(3)

```
%% 3-2 The initial locations of all the 100 mobile devices are decided uniformly in the 19-cell map.
MS_label  = randi(size(BS_X,2), 1, num_MS);
for i = 1:size(BS_X,2)
    num = sum(MS_label == i);
    if num > 0
        [x, y] = hexagon(side, BS_X(i), BS_Y(i), num, 0);
        X{i} = x;
        Y{i} = y;
    end
end
clear i;
clear x;
clear y;
clear num;
```

Introduction to Wireless and Mobile Networking
Hw3 – Report

B03901026 許凱傑

(4) Simulation

a Construct MS object

```
% initialize MS_randwalk objects
k = 1;
for i = 1:size(X,2)
    for j = 1:size(X{i},2)
        MS{k} = MS_randwalk(X{i}(j),Y{i}(j),0,0,0,i);
        [testX,testY,MS{k}] = MS{k}.update();
        k = k + 1;
    end
end
clear k;
clear i;
clear j;
```

b Draw the current MS scatter

```
figure;
hold on;
cellmap(side,BS_X,BS_Y,0,0,1,1);
for i = 1 : num_MS
    [MS_X, MS_Y]= MS{i}.getloc();
    text(MS_X, MS_Y, int2str(i), 'Color','b');
end
title('Figure B-2');
xlabel('Distance(m)'), ylabel('Distance(m)');
axis([-1.1*Xmax, 1.1*Xmax,-1.1*Ymax, 1.1*Ymax])
hold off;
```

c Simulation implemented by while loop

i Update location. If the MS goes beyond the border, see it goes to which surrounding cell map. Get the vector between the MS and the central BS of that cell map and move back to the original cell map.

```
for k = 1:num_MS
    % go to next location
    [testX, testY, MS{k}] = MS{k}.update();
    if ~inpolygon(testX, testY, borderX, borderY)
        for i = 1:6
            if inpolygon(testX, testY, outX{i}, outY{i})
                cell_label = i;
                break
            end
        end
        movetoX = testX - offsetX(cell_label);
        movetoY = testY - offsetY(cell_label);
        MS{k} = MS{k}.locate(movetoX,movetoY);
        %{
        fprintf('go beyond border\n');
        fprintf('MS(%d) goes to (%4.1f,%4.1f)\n',k,testX,testY);
        fprintf('MS(%d) goes to cell(%d)\n',k,cell_label);
        %}
    end
    % get transmitted power
    power(:,k) = MS{k}.power(BS_X, BS_Y, H_R, G_R);
end
```

Introduction to Wireless and Mobile Networking
Hw3 – Report

B03901026 許凱傑

ii Get SINR and see handover happen or not

```
% check handover
total = sum(power,2);
I = zeros(19,num_MS);
N = myThermalNoise(T,B);
for i = 1 : 19
    I(i,:) = total(i) - power(i,:);
end
BS_SINR = mySINR(power,I,N);
[M, maxlabel] = max(BS_SINR);
for i = 1 : num_MS
    [handover, oldlabel, MS{i} ] = MS{i}.handover(maxlabel(i));
    if handover == 1
        length = size(handover_msg,1);
        handover_msg(length+1, :) = {strcat(int2str(tf),'s'), oldlabel, maxlabel(i), i};
    end
end
```

iii Get output file and message

```
Table = cell2table(handover_msg,'VariableNames',{'Time' 'Source_cell_ID' 'Destination_cell_ID' 'MS_ID'} );
writetable(Table,'data.csv');

fprintf('The amount of total handover times is: %d\n', size(handover_msg,1));
```

2 Functions

(1) Number ↔ dB

```
% transfer to dB
function dB = todB(x)
dB = 10*log10(x);
end
```

```
% transfer from dB
function x = fromdB(dB)
x = 10.^(dB/10);
end
```

(2) Two-ray-ground model

```
% two-ray-ground-model
function G_d = G_two_ray_ground(H_t, H_r, d)
G_d = (H_t * H_r)^2 ./ (d .^ 4);
end
```

(3) Thermal Noise

```
% My Thermal Noise
function N_T = myThermalNoise(Temperature,Bandwidth)
k = physconst('Boltzmann');
N_T = k*Temperature*Bandwidth;
end
```

(4) SINR

```
% SINR in dB
function SINR = mySINR_dB(S, I, N)
SINR = 10*log10(S/(I+N));
end
```

Introduction to Wireless and Mobile Networking

Hw3 – Report

B03901026 許凱傑

(5) Hexagon

Can also randomly create num_MS MSs

By dividing each cell into 3 regions, and use 2 vectors combination to get each MS's location.

```
function [vectorX,vectorY] = hexagon(side,x0,y0,num_MS,draw)
    %side = side size;,(x0,y0) exagon center coordinates;
    L = linspace(0, 2*pi, 7);
    edgeX = side * cos(L)+x0;
    edgeY = side * sin(L)+y0;
    if draw
        plot(edgeX,edgeY,'r','Linewidth',1);
        scatter(x0,y0,'filled','g');
    end

    if num_MS >= 1
        ISD = side*sqrt(3);
        ai = [side,0.0];
        aj = [-side/2, ISD/2];
        ak = [-side/2, -ISD/2];
        temp  = randi(3,1,num_MS);
        tempx = rand(1,num_MS);
        tempy = rand(1,num_MS);
        for a =1:num_MS
            if temp(a) == 1
                x(a) = tempx(a)*ai(1) + tempy(a)*aj(1);
                y(a) = tempy(a)*aj(2);
            elseif temp(a) == 2
                x(a) = tempx(a)*ai(1) + tempy(a)*ak(1);
                y(a) = tempy(a)*ak(2) ;
            else temp(a) == 3
                x(a) = tempx(a)*aj(1) + tempy(a)*ak(1);
                y(a) = tempx(a)*aj(2) + tempy(a)*ak(2);
            end
        end;
        vectorX = x+x0;
        vectorY = y+y0;
        if draw
            scatter(vectorX,vectorY,10,'b','filled');
        end
    end;
```

(6) Hexagonborder

```
function [edgeX,edgeY] = hexagonborder(side,x0,y0, draw)
    %side = side size;,(x0,y0) exagon center coordinates;
    % draw(true) : draw the BS scatter and the subcell
    L = linspace(0, 2*pi,7);
    edgeX = side * cos(L)+x0;
    edgeY = side * sin(L)+y0;
    if draw == 1
        plot(edgeX,edgeY,'r','Linewidth',1);
        scatter(x0,y0,'filled','g');
    end
```

(7) Cellmap

Return the border of the 19cell-map

```
function [borderX, borderY] = cellmap (side, X, Y, centralX, centralY, label, draw)
    % label(true) : label BS number
    % side :length of the hexagon
    % X : relative BS x-location to central
    % Y : relative BS y-location to central
    % [centralX, centralY] : the location of central BS
    % [borderX, borderY] : the border of cellmap
    % draw(true) : draw the BS scatter and the subcell

    BS_X = X + centralX;
    BS_Y = Y + centralY;
    hold on;
    for i = 1:size(BS_X,2)
        [edgeX{i},edgeY{i}] = hexagonborder(side, BS_X(i), BS_Y(i), draw);
        if label == 1
            text(BS_X(i), BS_Y(i), int2str(i));
        end
        %text(centralX, centralY, '*');
        if i == 1
            borderX = edgeX{i};
            borderY = edgeY{i};
        else
            [borderX, borderY] = polybool('union', borderX, borderY, edgeX{i}, edgeY{i});
        end
    end
    clear i;
```

Introduction to Wireless and Mobile Networking

Hw3 – Report

B03901026 許凱傑

(8) MS_upade

```
function [v, theta ,t] = MS_update(minSpeed, maxSpeed, minT, maxT)
    v = unifrnd(minSpeed, maxSpeed);
    theta = unifrnd(1 , 2*pi);
    interval = maxT - minT;
    t = minT + unidrnd(interval);
```

3 Class MS_randwalk

(1) Properties

```
properties
    x
    y
    speed
    theta
    time
    label
end
```

```
properties (Constant = true, Hidden = true)
    % cannot change and don't show in the window
    minSpeed = 1;
    maxSpeed = 15;
    minT = 1;
    maxT = 6;
    P_MS   = 23 - 30;       % MS power = 23 dBm
    G_T_dB = 14;
    H_MS   = 1.5;           % height of mobile station
end
```

(2) Methods

a Constructor

```
function obj = MS_randwalk(x,y,speed,theta,time,label)
    % constructor
    if nargin == 6 % if having 6 i/p
        obj.x = x;
        obj.y = y;
        obj.speed = speed;
        obj.theta = theta;
        obj.time = time;
        obj.label = label;
    else
        obj.x = 0;
        obj.y = 0;
        obj.speed = 0;
        obj.theta = 0;
        obj.time = 0;
        obj.label = 0;
    end
end
```

b Update location and motion

```
function [x,y,obj] = update(obj)
    % update parameters
    obj.x = obj.x + obj.speed * cos(obj.theta);
    obj.y = obj.y + obj.speed * sin(obj.theta);
    obj.time = obj.time - 1;
    if obj.time <= 0
        [obj.speed, obj.theta, obj.time] = MS_update(obj.minSpeed, obj.maxSpeed,obj.minT, obj.maxT)
    end
    x = obj.x;
    y = obj.y;
end
```

Introduction to Wireless and Mobile Networking
Hw3 – Report

B03901026 許凱傑

c Directly locate MS

```
function obj = locate(obj,movetoX, movetoY)
    obj.x = movetoX;
    obj.y = movetoY;
end
```

d Get received power

```
function P_T = power(obj, x_b, y_b, H_R, G_R)
    % P_T = P_BS * G_T * G_C
    d_x = obj.x - x_b;
    d_y = obj.y - y_b;
    d_t = sqrt(d_x.^2 + d_y.^2);
    G_C = G_two_ray_ground(obj.H_MS, H_R, d_t);
    P_T = fromdB(obj.P_MS + obj.G_T_dB) * G_C *G_R;
end
```

e Get the location and Draw the scatter

```
function draw(obj)
  scatter(obj.x,obj.y,'b');
end
```

```
function [x,y] = getloc(obj)
    x = obj.x;
    y = obj.y;
end
```

f Check handover or not

```
function [handover, oldlabel, obj] = handover(obj, maxlabel)
    % check handover happen or not
    if obj.label == maxlabel
        handover = 0;
        oldlabel = obj.label;
    else
        handover = 1;
        oldlabel = obj.label;
        obj.label = maxlabel;
    end
end
```

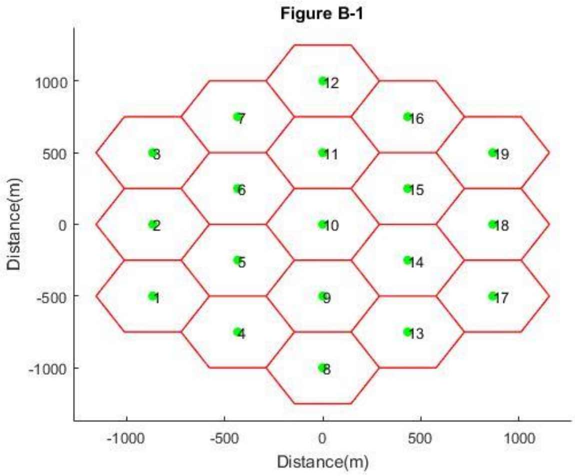
Introduction to Wireless and Mobile Networking

Hw3 – Report

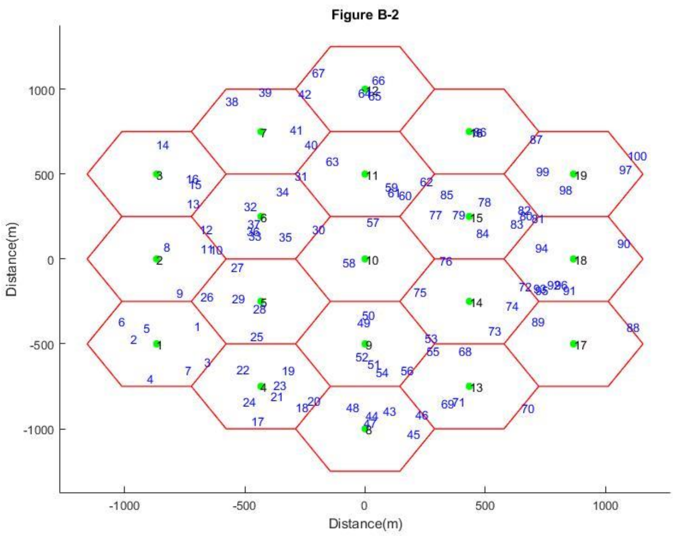
B03901026 許凱傑

II Report

1 *Please give a figure to describe how you arrange cell IDs to Fig. 1.*



2 *Please plot a map with all mobile devices in their initial location. Describe how you decide the initial location.*



See 1-(3) and 2-(5) to get the code and description . I use rand and randi as the base function to get the location of each MS.

3 *Based on B-1, please list all the time when the handoff event occurs and the related cell ID following the below format:*

Time	*Source cell ID*	*Destination cell ID*
3s	*1*	*2*

See the 'data.csv' to get this list.

Note: I add an extra column to show which MS happens handover.

4 *How many handoff events happen during the total simulation time?*

Average : 3750-3900.