

無線行動網路導論 HW3

B12901075 賴禹衡

第一題：

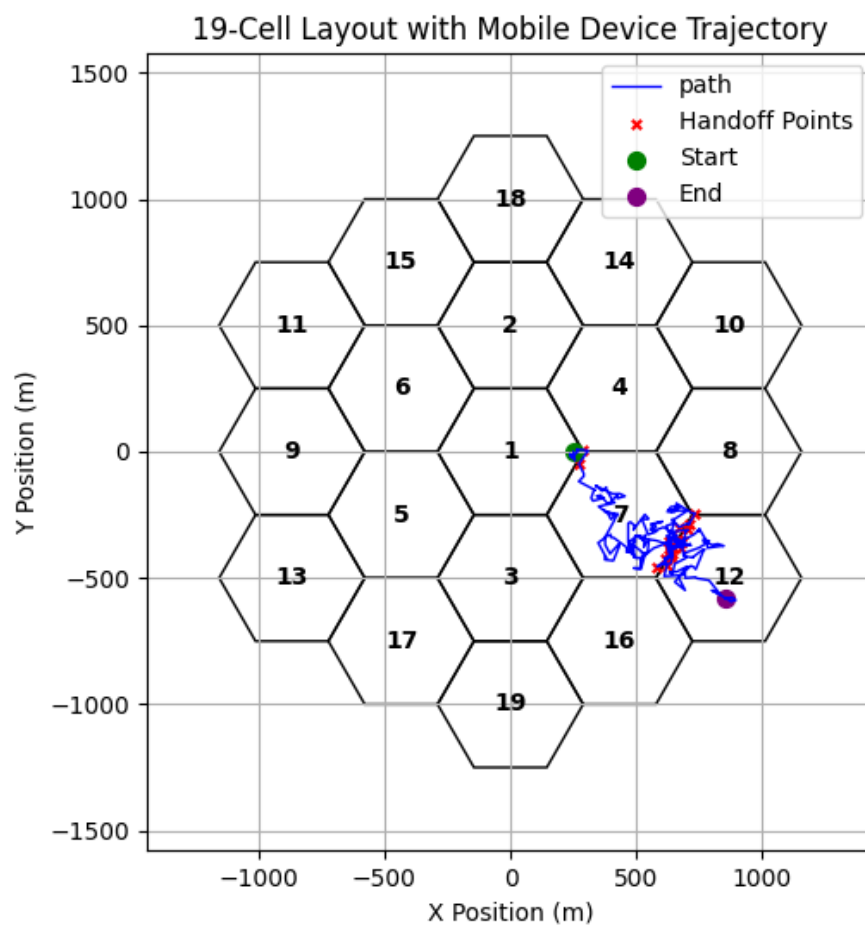
使用 Two-ray ground 模型

並以 coordinate 來判斷是否 handoff

而因為只有一個 device 所以會跑出邊界的機率極低因此沒有考慮

以本次為例有 22 次 handoff

結果儲存在 handoff_events.csv



Bonus:

使用 Two-ray ground 模型

並以 SINR 來判斷是否 handoff

我的找法是找到最大的 SINR 來判斷有沒有 handoff

也就是將 19 個 cell 都對每一個 md 做當下的 SINR 並判斷是否要 handoff

而程式碼則和 Hw2 滿像的

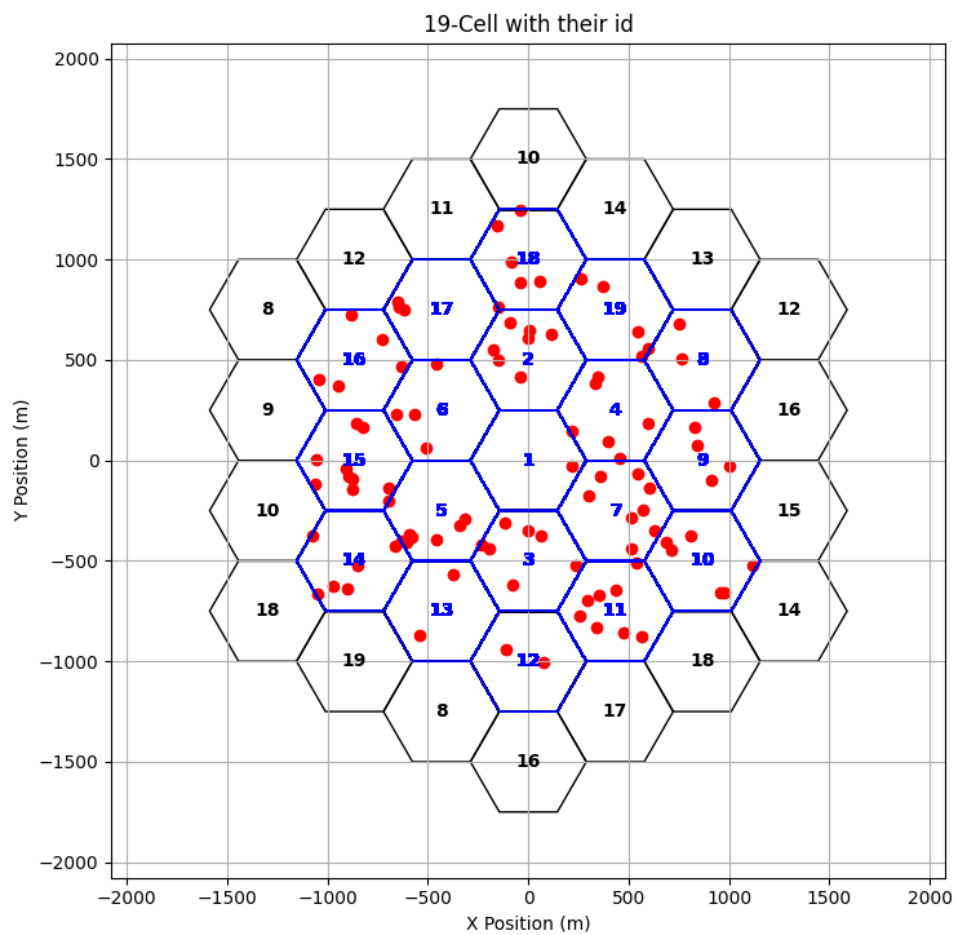
```
Title

1 def calculate_received_power(bs_position, md_position):
2     d = distance(bs_position, md_position)
3     P_received = P_m*(G_t*G_r*(h_bs**2)*(h_md**2))/(d**4)
4     return P_received
5
6 def calculate_SINR(md_position, serving_bs_id, bs_positions, devices):
7     P_signal = calculate_received_power(bs_positions[serving_bs_id - 1],
8     md_position)
9     P_interference = 0
10    for device in devices:
11        if device != md_position:
12            P_interference +=
13            calculate_received_power(bs_positions[serving_bs_id - 1], md_position) +
14            P_noise
15
16    SINR_linear = P_signal / (P_interference)
17    SINR_dB = 10 * math.log10(SINR_linear)
18    return SINR_linear
19
20 def checkconnectcell(bs_positions , md_position,devices):
21     tempSINR = -float('inf')
22     connectcell = 1
23     for q in range(len(bs_positions)):
24         if tempSINR < calculate_SINR(md_position,q+1,bs_positions,
25         devices ):
26             tempSINR = calculate_SINR(md_position,q+1,bs_positions,
27             devices )
28             connectcell = q+1
29     return connectcell
```

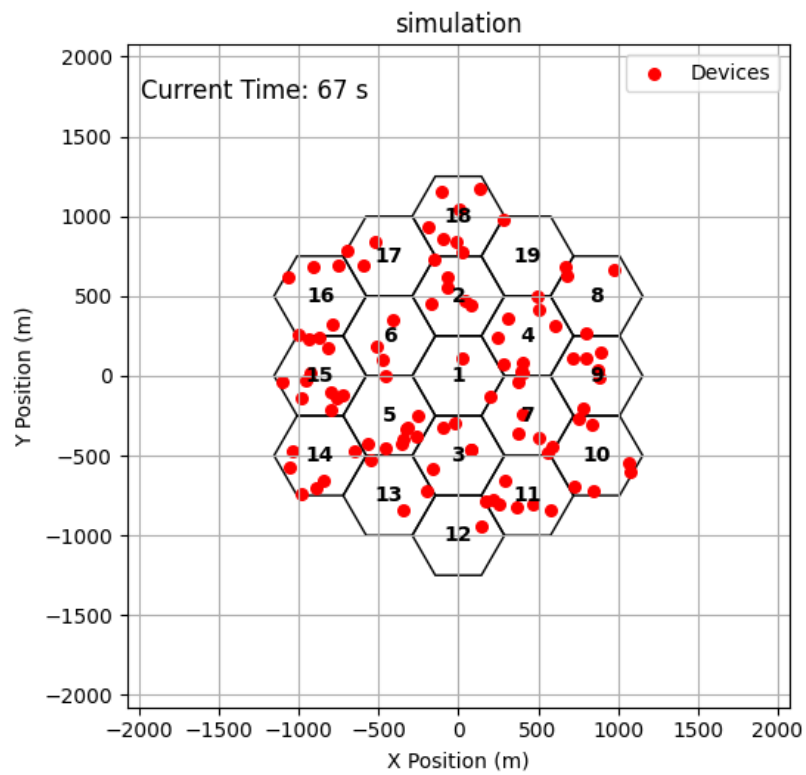
以上的程式碼大概的說明了我如何算出最大的 SINR cell

並用 matplotlib 來模擬 point 移動的過程
在外面圍上 18 個 cell 來處理跑出邊界的問題

以本次為例有 1914 次 handoff
結果儲存在 handoff_events_bonus.csv



黑色是外面圍上去的 cell 目的是判斷是否有跑出去



模擬示意圖

此外因為要求機率分佈是 `uniform` 的
因此我先 `uniform` 的選擇在哪一個 cell (`np.random.randint(1,19)`)
再處理在六邊形的哪一個位置 (參考 hw2)