

國立臺灣師範大學電機工程學系

碩士論文

指導教授：黃政吉 博士

5G 毫米波網路中用戶連線與資源配置演算法之設計

Designs of algorithms for client association and resource allocation
in mmWave 5G networks



研究生：邱昱瀚 撰

中華民國 一零八年七月

5G 毫米波網路中用戶連線與資源配置演算法之設計

學生：邱昱瀚

指導教授：黃政吉

國立臺灣師範大學電機工程系碩士班

摘要

5G毫米波網路可透過使用波束成形(beamforming)技術、與小細胞接入點(small cell access points, AP)的密集部署，以提供Gbps等級的速率。在本論文中，我們考慮用戶連線與通道資源分配的問題。該問題可以表示為整數非線性最佳化。為了解決該問題，我們首先利用對偶分解(dual decomposition)法來獲得基準場景的集中式(centralized)演算法。在基準場景中，並沒有考慮用戶端需求。接著，我們將用戶本身的需求分別考慮在限制中或作為目標中的權重，然後提出遞迴式與分散式(distributed)演算法。數值結果顯示我們提出的兼具用戶需求考量的演算法在吞吐量方面可明顯地優於基準場景之演算法。

關鍵字：對偶分解、毫米波、分散式演算法、集中式演算法、用戶需求。

Designs of algorithms for client association and resource allocation in mmWave 5G networks

Student : Ciou, Yu-Han

Advisor : Dr. Huang, Jeng-Ji

Department of Electrical Engineering
National Taiwan Normal University

ABSTRACT

A 5G millimeter wave network is able to deliver speeds of Gbps through the use of beamforming and the dense deployment of small cell access points (APs). In this thesis, we consider the problem of the associations of clients and allocations of channel resources. The problem can thus be formulated into an integer nonlinear optimization. To solve the problem, we first utilize a dual decomposition method to obtain a centralized algorithm for a baseline scenario. In the baseline scenario, no client demands are considered. Next, demands of clients are taken into account either in the constraints or as weights in the objective, and a distributed algorithm is then proposed. Numerical results demonstrate that our proposed algorithm for scenarios with client demands can substantially outperform the one for scenarios without them.

Keywords—dual decomposition, mmW, centralized algorithm, distributed algorithm, demand.

誌 謝

在進行對 5G 的研究時，最需要感謝的是我的指導教授黃政吉博士。在每周的會議上，老師會用不斷提問，與我討論問題，引導我主動去思考，使我知道下一步的方向要怎做，課題要怎麼進行下去等。

老師常常告誡我，對任何細節都要認真的想過，並且要要有獨立解決問題的能力，不論在現在的研究，還是在未來的工作，這都是非常重要的能力。雖然這使我常常陷入迷惘，但追隨老師的引導，慢慢的我能主動解決原先不知的問題，研究也因此成型，在此必須致上最深的感謝，感謝老師願意陪我繞許多遠路，只為了讓我能獨立思考，解決問題。

接著要感謝的是跟我同一個實驗室同學陳庭玉。在研究上，有時候遇到無法理解的公式，她會進行耐心的講解。而且當系上許多事情都是她幫我忙，才能順利解決，她也時常注意到我所沒注意的細節，使我不至於粗心犯錯。

最後要感謝的是我的家人，在求學時依然在我身邊陪伴我，關心我，並給我最大的支持與鼓勵。

邱昱瀚 謹致於
中華民國 一零八年七月

目 錄

中文摘要	i
英文摘要	ii
誌 謝	iii
目 錄	ivv
表 目 錄	vii
圖 目 錄	viii
第一章 緒論	1
1.1 5G 的背景	1
1.2 毫米波(mmW)通訊	3
1.3 路徑損耗	4
1.4 公平性原則	4
1.5 論文動機	4
1.6 章節安排	5
第二章 重要相關文獻與相關背景知識	6
2.1 [7]	6
2.2 利用距離推導訊雜比[5]	8
2.3 [6]	12
2.4 對偶分解法	15
2.5 資源分配與公平性[15]	19
第三章 分析方法	22
3.1 連線最佳化	22
3.2 集中式演算法	25
3.3 用戶頻寬的需求	35
3.4 用戶之權重	39
第四章 數值結果與討論	44
4.1 實驗環境	44
4.2 連線最佳化問題的結果與討論	45
4.3 加入需求的限制	50
4.4 模擬加入權重的演算法	53
第五章 結論	58
附 錄 A 算幾不等式	59
附 錄 B 斯萊特條件與強對偶性	61
附 錄 C 權重轉換	62
參 考 文 獻	63



表 目 錄

表 1：基於紐約市大規模數據之參數統計模型[5]。	10
表 2：模擬所採用之參數	44



圖 目 錄

圖 1- 1：5G 應用場景[14]。	3
圖 2- 1：Google 地圖，其中藍線顯示用戶移動路徑。	8
圖 2- 2：鏈路狀態機率與傳輸距離關係圖[5]。	9
圖 2- 3：AP 之資源配置示意圖示[6]。	13
圖 2- 4：AP 與用戶連線結果圖[6]。	14
圖 2- 5：自然對數的函數圖像。	20
圖 3- 1：用戶與 AP 之間的連線與資源分配範例；(a)所有連線可能性；(b) 連線結果；(c)各個 AP 的資源分配。	23
圖 3- 2：自然對數函數。	24
圖 3- 3：演算法 1。	30
圖 3- 4：(a)小型測試範例；(b)窮舉法最佳解之連線。	32
圖 3- 5：小型測試範例在演算法 1 之運算過程。	33
圖 3- 6：演算法 2。	37
圖 3- 7：演算法 3。	43
圖 4- 1：AP 與用戶的位置分布結果圖。	45
圖 4- 2：AP 與用戶的連線結果圖。其中，不同的連線顏色代表用戶與對 應顏色的 AP 進行連線。	46
圖 4- 3： λ_k 的演進過程。	48
圖 4- 4： n_k 的演進過程。	48
圖 4- 5： $x_k = \sum_{i \in C} x_{ik}$ 的演進過程。	49
圖 4- 6：當不同需求限制時，演算法 2 之增益。	50
圖 4- 7：不同的群組 1 用戶數量 $ G1 $ ，演算法 2 相對於演算法 1 之增益 G， 其中 Range=0.1。	52
圖 4- 8：AP 與用戶的連線分布圖；(a)無權重時；(b)有權重時。	53
圖 4- 9：將兩個結果重疊之連線分布圖。	54
圖 4- 10： n_k 的演進過程；(a)有權重時；(b)無權重時。	55
圖 4- 11： λ_k 的演進過程；(a)有權重時；(b)無權重時。	57

第一章 緒論

在本章中，首先在 1.1 節介紹 5G 之背景，其次在 1.2 與 1.3 節分別說明毫米波通訊與路徑損失，然後在 1.4 節討論公平性原則，最後在 1.5 節陳述本論文之研究動機。1.6 節則是章節安排。

1.1 5G 的背景

隨著時代的發展，人們對高數據速率服務的需求不斷增長。為此，5G 無線通信被提出來應對不斷增長的需求。

對於 5G 無線通信來說，其技術創新的特點如下：

- 毫米波 (mmW) 通訊：毫米波是指波長在毫米數量級 (order of magnitude) 的電磁波，其頻率一般在 6 GHz 以上。由於目前無線網路所使用的 5 GHz 以下頻段已經壅擠，因此如果要使用新的頻譜資源，須使用毫米波技術 [1]；
- 微基站 (small cell base station) 技術：由於毫米波通訊頻率高，波長短，導致其主要問題為高穿透損耗 (penetration loss)。為此，若使用微基站技術 [2]–[7]，可使得功耗大幅度降低。
- 大規模多輸入多輸出 (Multiple-Input Multiple-Output, Massive MIMO) 相控陣列天線：使用大量天線並進行同步處理，使小基站能向更多用戶發送和接收訊號，以大幅度提升網路容量 [8]；
- 波束自適應和波束賦形 (Beamforming)：使用大規模多輸入多輸

出技術，可控制訊號使其能對準用戶，具有方向性[9]；

- 設備-設備通訊技術(Device to Device, D2D):使用 5G 技術時，可無須透過基地台傳送訊號，而是透過相鄰的用戶直接連線，以降低基地台的通道資源[10]。

相對於4G,5G具有許多優勢，其中包含極高的資料傳輸速率達10 Gbit/s、較低的延遲、能更快回應等，詳述如下：

- 延遲(delay)低：4G由於先天的限制，會有10 ms以上的延遲，這導致無法適用於一些重要場合，如自動駕駛、手術等。5G卻可以達到5 ms以下的延遲[11]；
- 資料傳輸速率更高：隨著物聯網發展、照片辨識率的提升等因素，對高數據速率服務的需求愈來愈高。為此，需要5G來應對其需求[12]；
- 低功耗：由於毫米波技術的缺點，需使用微基站技術，使得用戶功率消耗大幅度下降[13]。

因此，5G可利用這些優勢發展出許多應用服務，例如：移動醫療(Mobile Health, M-Health)、自動駕駛、車聯網(Internet of Vehicles, IoV)、物聯網(Internet of Things, IoT)等，如圖1-2。

在本篇論文，我們主要探討5G的毫米波技術與波束賦形技術，詳見於2.1與2.3節。

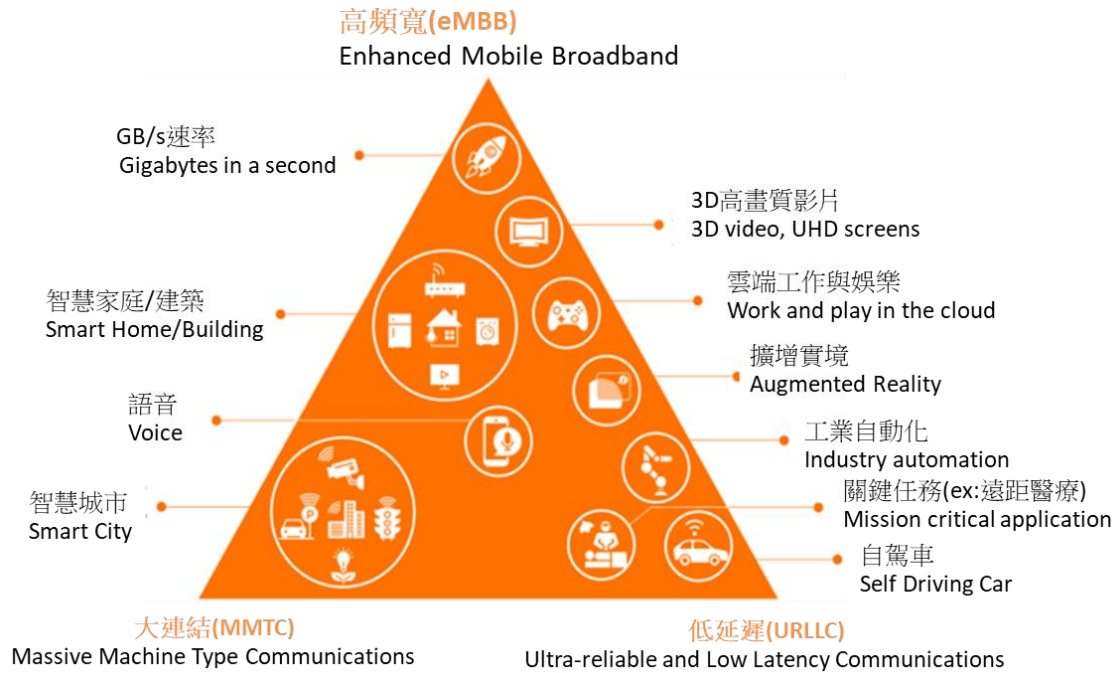


圖 1-1：5G 應用場景[14]。

1.2 毫米波(mmW)通訊

毫米波通訊是5G技術中，資料傳輸速率高於4G的原因之一。其中，毫米波的頻段大約在30-300 GHz[11]，目前主要運用在28 GHz。而在28 GHz頻段，5G網路的可用頻寬達1 GHz [10]。

相對於4G微波通訊(Microwave, μW)來說，毫米波通訊具有嚴重的路徑損耗(path-loss)以及高穿透損耗(penetration loss)。為此，可利用筆型波束(pencil beam)[6],[7]，以克服上述的缺點，詳見於2.1與2.3節。由於筆型波束極少遭受干擾，使得干擾並非毫米波技術的主要議題，這與4G微波網路非常不同。第三章中，我們針對毫米波的通訊特性，提出新的連線方法，並整理成連線最佳化問題；除此之外，我們並提出對應的演算法，如第3章所述。

1.3 路徑損耗

在本篇論文中，我們需要測定訊號雜訊功率比(signal-to-noise ratio, SNR)，簡稱訊雜比，以作為連線最佳化問題的參數。為此，我們運用[5]中透過實地測量所獲得的統計模型，其中包含：以用戶與 AP 間之距離為變數的路徑損耗(path loss)函數、不同頻率與不同鏈路狀態所對應之參數等。如此一來，若用戶與接入點(access point, AP)的位置為已知，便可計算出訊雜比SNR，詳細計算如2.2節所述。

1.4 公平性原則

由於5G毫米波網路與4G網路考慮的主要問題不同，因此，須提出的新的連線最佳化問題與演算法。為此，在 2.4 節中，我們介紹了幾種常見的公平性原則，如具權值的比例公平性(Weighted proportional fairness)、最小潛在延遲公平性(minimum potential delay fairness)、與最大最小公平性(max-min fairness)等，可作為連線問題的考量[15]。在本篇論文中，主要使用比例公平性作為連線問題的公平性原則。

1.5 論文動機

面對不斷增長的高數據速率服務之需求，在5G網路中，將使用毫米波技術，以提高資料傳輸速率。在本論文中，如 1.2 節所述，我們提出新的連線方式，能顯著地提高系統吞吐量與公平性。

尤其，我們考慮用戶本身的需求，以免造成不必要的資源浪費，並可進一步提高網路的吞吐量。

在本篇論文中，我們的主要貢獻總結如下：

- 提出新的連線最佳化問題，做為用戶連線的方法；
- 類似[6]，將最佳化問題改寫成多維分配問題，並使用對偶分解(dual decomposition)法進行求解；
- 類似[6]，針對對偶分解(dual decomposition)法，提出集中式(centralized)與分散式(distributed)的演算法，以解決多維分配問題；
- 考慮用戶本身的需求，在最佳化問題加入需求限制，提出一種遞迴的演算法；
- 使用權重以對應用戶的需求，並針對新的最佳化問題，提出分散式演算法。

1.6 章節安排

在本篇論文中，其餘章節的安排如下。在第二章中，我們介紹相關文獻，以說明求解連線最佳化問題時所需要的基礎相關。在第三章中，我們首先陳述主要探討的連線最佳化問題；接著，針對該連線最佳化問題，提出集中式(centralized)與分散式(distributed)演算法；最後，我們考慮用戶需求，分別加入需求限制與權重後，並整理為對應之連線最佳化問題。在第四章中，我們利用電腦模擬所提出的演算法，並展現數值結果與討論。最後，我們的結論放在第五章。

第二章 重要相關文獻與相關背景知識

我們首先介紹一些背景知識，以成為設計基於公平性與用戶需求演算法之基礎。為此，我們介紹幾篇重要文獻，例如：[6],[7]，以及一些作法；例如：對偶分解(dual decomposition)等。

2.1 [7]

在這篇論文中，提出5G毫米波(mmW)網路中，移動性已知的用戶連線策略，以改善使用傳統連線策略利用最大接收訊號強度(received signal strength, RSS)的缺點。由於5G的優點，在於使用毫米波通信與網路密集化。但是，相對於4G來說，卻有較高的路徑損失(path-loss)、與對阻塞的高敏感性(blockage)。為了解決上述的問題，毫米波通信在小空間放置大量天線元件，以形成窄波束，或稱為筆型波束(pencil beam)，以具有高度的方向性。藉助這種方向性，在毫米波頻帶所建立的鏈路範圍可達到約 200 公尺。此外，網路密集化亦有助於毫米波網路。主要因為網路密集化能提高小基站(sBs)使用直視波進行傳輸(line of sight, LOS)的機會。

由於毫米波網路使用筆型波束，干擾將會大幅度降低，大多情況只需考慮雜訊。因此，以最大接收訊號強度為基準的連線策略，將不再適用於毫米波網路。所謂的最大接收訊號強度為基準的連線策略，是用戶會與能提供最大接收訊號強度的小基站(small base station, sBs)進行連線。對於5G網路來說，基於最大接收訊號強度的連線有下列缺點：

- 首先，該連線策略會導致小基站的連線用戶數量不均衡，也就是所謂負載不均衡。當負載不平衡時，在用戶密集地區，每位用戶的可用資源將十分有限；而在用戶稀疏地區，將會有過多資源，造成浪費。如此，將嚴重影響毫米波網路的公平性(fairness);
- 另外，該連線策略可能導致用戶在相鄰的小基站之間，進行過度頻繁的切換。由於用戶不斷地重新連線，形成沉重的開銷(overhead)，而導致額外的資源浪費。

於是，該論文提出一個移動性已知的用戶連線策略，以適用於毫米波網路的環境，並可改善上述最大接收訊號強度為基礎之連線策略的缺點。該論文所設計的用戶連線策略，採用時槽的方式，讓小基站服務連線的用戶，其中時槽為固定長度 T_s 。該用戶連線策略展現下列特點:

- 可追蹤網絡拓撲的動態變化，以及用戶移動所導致通道條件的改變；
- 考慮小基站的負載，避免讓用戶與擁塞的小基站連線。如此，對於網絡的整體公平性，可產生積極影響；
- 克服用戶在小基站間過於頻繁的切換，使得用戶不需要頻繁的重新連線，而可減少不必要的資源浪費；
- 考慮毫米波通信的特性，如方向性、阻塞敏感性、以及非直視波傳輸；
- 該作法為全分散式。

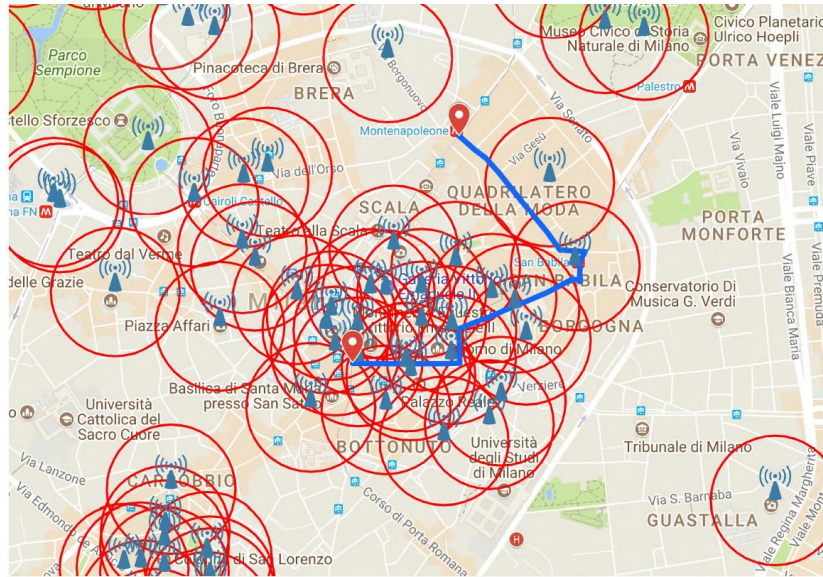


圖 2-1 : Google 地圖，其中藍線顯示用戶移動路徑。

圖2-1為該論文模擬時所用的路徑，地點是在意大利米蘭市中心。其中，該論文把目前wifi基站位置視為5G的小基站位置。透過模擬結果，對比於最大接收訊號強度為基礎之連線策略，該用戶連線策略的各種特點，整理如下：

- 用戶移動速度愈快，該用戶連線策略的效能愈好；
- 負載愈不均衡，該用戶連線策略的效能愈好；

2.2 利用距離推導訊雜比[5]

由於本篇論文探討連線的最佳化問題，因此需要計算用戶*i*與AP *k*之間的連線速率 r_{ik} ，以作為連線決策之依據。假設用戶與AP的距離 d_{ik} 是已知的。如此，便可透過下列的計算公式，計算訊號雜訊功率比(signal-to-noise ratio, SNR)，或簡稱訊雜比。然後，再利用香濃定理(Shannon's theorem)，計算 r_{ik} 。

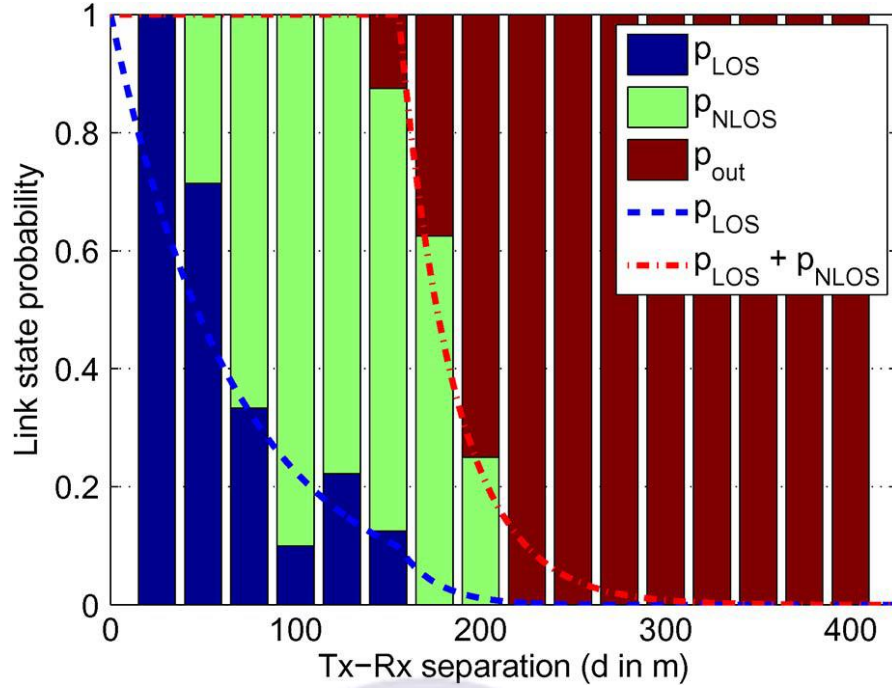


圖2-2：鏈路狀態機率與傳輸距離關係圖[5]。

2.2.1 狀態機率分佈

根據3GPP的規範，AP與用戶之間的鏈路有2個狀態：直視波傳輸(line of sight, LOS)與非直視波傳輸(non-line of sight, NLOS)狀態；狀態的機率為距離 d_{ik} 的函數，如圖2-2所示。在[5]中，毫米波網路另增加一個中斷(outage)狀態。因此，AP與用戶間的鏈路有3個狀態：直視波傳輸、非直視波傳輸、與中斷。透過多次實際實驗所產生的統計模型，可得3種狀態機率函數如下(2.1)：

$$p_{out}(d) = \max(0, 1 - e^{-a_{out}d + b_{out}}), \quad (2.1a)$$

$$p_{LOS}(d) = (1 - p_{out}(d))e^{-a_{los}d}, \quad (2.1b)$$

$$p_{NLOS}(d) = 1 - p_{out}(d) - p_{LOS}(d). \quad (2.1c)$$

表 1：基於紐約市大規模數據之參數統計模型[5]

變數	模型	模型參數值	
		28 GHz	73 GHz
路徑損失 PL 與對 數陰影 ξ	$PL = \alpha + 10\beta \ln d + \xi$ [dB] $\xi \sim \mathcal{N}(0, \sigma^2)$, d 為距離	非直視性: $\alpha = 72.0, \beta = 2.92,$ $\sigma = 8.7$ dB 直視性: $\alpha = 61.4, \beta = 2,$ $\sigma = 5.8$ dB	非直視性: $\alpha = 86.6, \beta = 2.45,$ $\sigma = 8.0$ dB (1) 直視性: $\alpha = 82.7, \beta = 2.69,$ $\sigma = 7.7$ dB (2) 直視性: $\alpha = 69.8, \beta = 2,$ $\sigma = 5.8$ dB
非直視波 -直視波- 中斷機率	看 (4.1)	$1/a_{out} = 30.0$ m, $b_{out} = 5.2,$ $1/a_{los} = 67.1$ m	

(1) 2 m-RX-高度數據和 4.06 m-RX-高度數據的參數組合

(2) 僅限 2 m-RX-height 數據的參數

其中，參數 a_{out} ， a_{los} ， b_{out} 之數值皆透過實驗結果統計而得，如表 1。根據 (2.1)，我們可分別計算 d_{ik} 之中斷機率 p_{out} ，直視波傳輸機率 p_{LOS} 與非直視波傳輸機率 p_{NLOS} ，其結果可如圖 2-4 中藍色與紅色虛線所分割之三種區域。

我們可以從圖 2-4 看出，隨著距離 d_{ik} 增加，直視波傳輸機率愈小，而非直視波傳輸機率則愈多。直到距離 d_{ik} 達到 200 m 時，中斷機率為百分之百。因此，狀態機率會受到 d_{ik} 影響。根據 d_{ik} 所對應之狀態機率，可利用均勻分布之隨機變數，以決定 AP 與用戶之間的鏈路狀態。

2.2.2 路徑損失

在決定AP與用戶間的鏈路狀態之後，我們便可根據表1的參數值，計算AP與用戶間鏈路的路徑損耗PL。然後，利用路徑損耗PL，以計算訊號雜訊功率比SNR。路徑損耗PL的標準線性模型如下：

$$PL [dB] = \alpha + 10\beta \ln d + \xi, \quad \xi \sim \mathcal{N}(0, \sigma^2). \quad (2.2)$$

其中， d 為AP與用戶間的距離，其單位為公尺。參數 α 與 β 分別為利用最小平方法所計算之截距與斜率(least square fits)， σ^2 則是對數常態遮蔽效應變異數(lognormal shadowing variance)。須注意在確定AP與用戶之間連線狀態的情況下， α 、 β 、 σ^2 之值都可以從表1取得。

取得PL之後，我們可以根據下式，以dB的方式計算接收功率 P_{RX} ：

$$PL = P_{TX} - P_{RX} + G_{TX} + G_{RX}, \quad (2.3)$$

其中， P_{TX} 為總傳輸功率，單位為dBm； P_{RX} 為接收功率； G_{TX} 、 G_{RX} 則是天線的增益。在[5]中， $P_{TX} = 30$ dBm， $G_{TX} = G_{RX} = 24.5$ dBi。

最後，我們仍以dB的方式，利用 P_{RX} 求得訊雜比如下：

$$SNR = P_{RX} - P_{n_dB}, \quad (2.4a)$$

$$P_{n_dB} = P_{noise} + NF [dB]. \quad (2.4b)$$

其中，雜訊功率頻譜密度 $P_{noise} = -174 \text{ dBm/Hz}$ ，而接收端的噪音因數(noise figure, NF)為6 dB[6]。

2.3 [6]

這篇論文設計分散式的演算法，以解決毫米波(mmW)網路的用戶連線與通道資源分配。用戶可透過中繼(relay)，或直接與AP建立連線，並考慮資源分配的公平性。在5G無線通信中，毫米波網路是一項可以實現高數據速率(high data rate)無線通訊的重要技術。然而，毫米波網路主要的挑戰包含高路徑損耗(path-loss)、對阻塞的高敏感性(blockage)、與由於傳送端與接收端的波束並未對準所造成的耳聾(deafness)。

為了克服上述的挑戰，在該論文中，提出兩種方法：

- 首先，針對高路徑損耗與對阻塞的高敏感性，可在小空間中放置大量的天線元件，以形成窄波束，或稱為筆型波束(pencil beam)。這些窄波束具有方向性，可以大幅度減少干擾(interference)，並且提供高天線增益(antenna gain)。須注意，由於干擾的影響大幅度減少，毫米波網路通常只需考慮雜訊(noise)。
- 另外，可加入中繼以建立有效率的連線。這是因為中繼技術可提供訊號較佳的連線(robust connection)、負載均衡(load balancing)、覆蓋範圍擴展(coverage extension)、室內外覆蓋(indoor-outdoor coverage)、高效的移動性管理(mobility management)、以及平穩的換手操作(smooth handover operation)，使得連線具備更一致的服務品質。

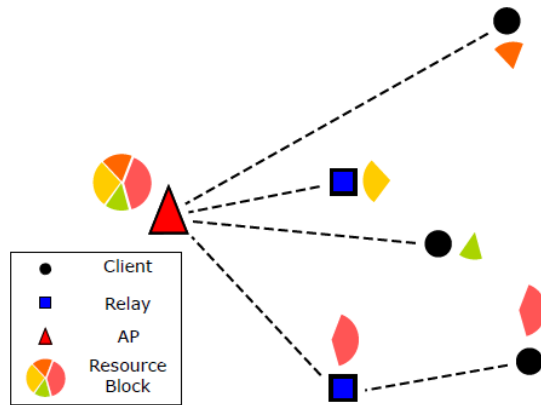


圖2-3：AP之資源配置示意圖示[6]。

這篇論文研究了毫米波網路的聯合連線與中繼問題，並設計為最佳化問題進行求解，其過程如下：

- 首先，提出一個聯合(joint)連線與中繼的最佳化問題，其中包含對數效用(logarithmic utility)、AP的資源分配、客戶連線的選擇、中繼的選擇、與不完整的通道狀態訊息(imperfect channel state information)；
- 接著，由於上述的最佳化問題無法直接求解。於是，該論文將其改寫為多維分配問題，並提出對偶分解法(dual decomposition)進行求解；
- 最後，建立分散式拍賣演算法，以解決上述的多維分配問題。

關於用戶連線與通道資源分配，該論文提出若干的限制，如圖2-3，其中包含：

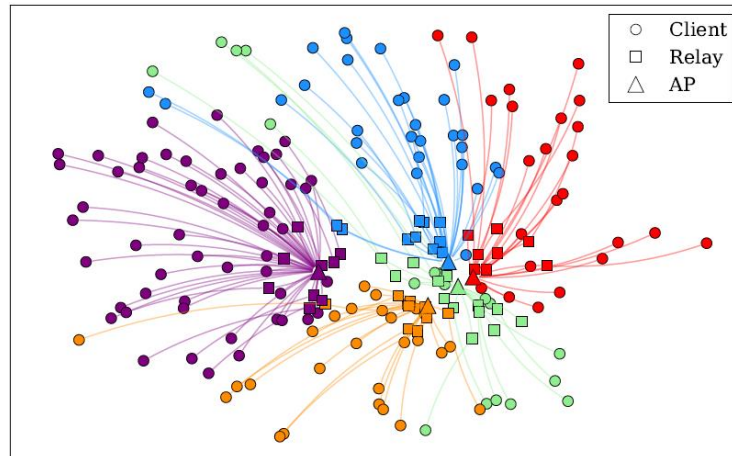


圖 2-4: AP 與用戶連線結果圖[6]。

- 一個中繼只能協助1個用戶與AP進行連線；
- 1個用戶(或中繼)只能與一個AP進行連線；
- 一個AP分配給用戶的通道資源比例，其總和不能大於1，如圖2-3所示；
- 當用戶透過中繼連線時，該用戶所分配的通道資源比例，與該中繼所分配的通道資源比例相同。

該論文之通道資源分配，可以圖2-3為例，其中包含3個用戶、2個中繼；用戶與中繼旁邊的扇形區域，代表AP所分配的通道資源比例。關於分配的通道資源比例，其所代表的意義為：用戶(client)或中繼(relay)與AP的連線時間的比例。須注意若用戶透過中繼與AP連線，該用戶所分配的通道資源比例，如圖2-3右下角用戶所對應的扇形區域，與中繼所分配的通道資源比例，也就是2-3下方中繼所對應的扇形區域相同。

圖2-4為該論文連線最佳化演算法的執行結果。可以看出大部分用戶會跟較近的AP進行連線。這是因為用戶與AP之間的連線速率較快。然而，並非所有用戶都選擇與速率最高的AP進行連線。少部分用戶會選擇較遠但負載

較小的 AP 進行連線。

2.4 對偶分解法

在本篇論文中，我們考慮毫米波 mmW 網路中，用戶連線的最佳化問題。如稍後第三章將介紹，該問題可利用對偶分解法(dual decomposition)以尋求最佳解。因此，在本節中，我們介紹該方法，以及相關的證明。

假設最佳化問題之目標函數可表示如下：

$$\max_{x,y,z} f(x,z) + g(y,z). \quad (2.5)$$

可發現 z 為共同變數，而 x 、 y 則分別為 f 、 g 個別的變數。針對(2.5)，我們可以利用對偶分解法，以求得最佳解。其做法如下所述。

首先，我們可以另加入限制 $z = z_1 = z_2$ ，(2.5)可等效為(2.6):

$$\max_{x,y,z_1,z_2} f(x,z_1) + g(y,z_2) \quad (2.6a)$$

$$\text{s. t.} \quad z_1 = z_2. \quad (2.6b)$$

利用拉格朗日乘數法(Lagrangian multiplier)，可將(2.6)的目標與限制，改寫為拉格朗日函數(Lagrangian function):

$$\mathcal{L}(x, y, z_1, z_2, \lambda) = f(x, z_1) + g(y, z_2) + \lambda(z_1 - z_2), \quad (2.7)$$

接著，它的拉格朗日對偶方程式(Lagrangian dual function)可表示為：

$$\begin{aligned}
\mathcal{D}(\lambda) &= \max_{x,y,z_1,z_2} \mathcal{L}(x,y,z_1,z_2,\lambda) \\
&= \max_{x,y,z_1,z_2} [f(x,z_1) + \lambda z_1] + [g(y,z_2) - \lambda z_2] \\
&= \max_{x,z_1} [f(x,z_1) + \lambda z_1] + \max_{y,z_2} [g(y,z_2) - \lambda z_2] \\
&= \max_{x,z} [f(x,z) + \lambda z] + \max_{y,z} [g(y,z) - \lambda z], \quad (2.8)
\end{aligned}$$

稍後可知，(2.8)為(2.5)對偶問題(dual problem)之目標函數。令 p^* 為(2.5)的最佳目標值，即 $p^* = \max_{x,y,z} f(x,z) + g(y,z)$ 。我們可以繼續由(2.8)獲得(2.9):

$$\mathcal{D}(\lambda) = \max_{x,z} [f(x,z) + \lambda z] + \max_{y,z} [g(y,z) - \lambda z] \quad (2.9a)$$

$$\geq \max_{x,z} [f(x,z) + \lambda z + g(y,z) - \lambda z] \quad (2.9b)$$

$$= \max_{x,y,z} f(x,z) + g(y,z) = p^*. \quad (2.9c)$$

可以發現 p^* 是 $\mathcal{D}(\lambda)$ 的下界(lower bound)，即 $\mathcal{D}(\lambda) \geq p^*$ 。稍後，我們將計算 $\min_{\lambda} \mathcal{D}(\lambda)$ ，以使得該目標值，盡可能接近 p^* 。關於(2.9a) \geq (2.9b)之證明如下。

定理 2.1 z 為一實數變數，則： $\max_z f(z) + \max_z g(z) \geq \max_z (f(z) + g(z))$ 。

Proof: 假設 $\max_z f(z) = f(z_1)$ 、 $\max_z g(z) = g(z_2)$ 、 $\max_z (f(z) + g(z)) = f(z_3) + g(z_3)$ 。根據前兩個假設，可得：

$$f(z_1) \geq f(z_3), \quad (2.10a)$$

$$g(z_2) \geq g(z_3). \quad (2.10b)$$

將(2.10a)與(2.10b)相加，可得：

$$f(z_1) + g(z_2) \geq f(z_3) + g(z_3), \quad (2.11)$$

(2.11)可因此改寫為:

$$\max_z f(z) + \max_z g(z) \geq \max_z (f(z) + g(z)). \quad (2.12)$$

如前所述，若 p^* 為(2.5)的最佳目標值， $\mathcal{D}(\lambda)$ 則為其對偶問題的目標。由上述證明可以看出， $\mathcal{D}(\lambda) \geq p^*$ 。當 p^* 無法由(2.5)直接求解時，則以 $\min_{\lambda} \mathcal{D}(\lambda)$ 作為 p^* 之近似最佳解。因此，(2.5)的對偶問題可表示為:

$$\min_{\lambda} \mathcal{D}(\lambda) = \min_{\lambda} \{ \max_{x,z} [f(x,z) + \lambda z] + \max_{y,z} [g(y,z) - \lambda z] \}. \quad (2.13)$$

接著，將(2.13)進行拆分並進行求解，也就是改寫為一個主問題與兩個子問題:

主問題:

$$\min_{\lambda} \mathcal{D}(\lambda), \quad (2.14a)$$

子問題:

$$\max_{x,z} [f(x,z) + \lambda z], \quad (2.14b)$$

$$\max_{y,z} [g(y,z) - \lambda z]. \quad (2.14c)$$

(2.13)的求解過程需要不斷地更新 λ ，主要包含下列兩個步驟:

- 首先，針對目前的 λ ，分別計算(2.14b)與(2.14c)對應之最佳解；
- 接著變換 λ ，變換幅度與 $\mathcal{D}(\lambda)$ 對 λ 作偏微分之斜率 $\frac{\partial \mathcal{D}}{\partial \lambda}$ 成正比。

令 $(x^*, z_1^*) = \arg \max_{x,z} [f(x, z) + \lambda z]$ ， $(y^*, z_2^*) = \arg \max_{y,z} [g(y, z) - \lambda z]$ ， $\frac{\partial \mathcal{D}}{\partial \lambda} = z_1^* - z_2^*$ 。若 $z^* = z_1^* = z_2^*$ ，由於斜率 $\frac{\partial \mathcal{D}}{\partial \lambda} = 0$ ，(2.14a)有最小值，也就是(2.13)之近似最佳解；相反地，若 $z_1^* \neq z_2^*$ ，或斜率 $\frac{\partial \mathcal{D}}{\partial \lambda} \neq 0$ 時， λ 將更換為新的數值 λ' ：

$$\lambda' = \lambda - \delta \times \frac{\partial \mathcal{D}}{\partial \lambda} = \lambda - \delta \times (z_1^* - z_2^*). \quad (2.15)$$

其中， δ 為基本變換幅度。

須注意，當 $\frac{\partial \mathcal{D}}{\partial \lambda} = 0$ ，也就是 $z^* = z_1^* = z_2^*$ 時，可得(2.13)之最佳解，且其目標值與原始問題(2.5)之最佳目標值相同。其理由如下：

$$\begin{aligned} \mathcal{D}(\lambda) &= \max_{x,z} [f(x, z) + \lambda z] + \max_{y,z} [g(y, z) - \lambda z] \\ &= [f(x^*, z_1^*) + \lambda z_1^*] + [g(y^*, z_2^*) - \lambda z_2^*] \quad (\text{如(2.15)中之定義}) \\ &= f(x^*, z^*) + g(y^*, z^*). \end{aligned} \quad (2.16)$$

如前所述， $\mathcal{D}(\lambda) \geq p^*$ ，且根據上式， $f(x^*, z^*) + g(y^*, z^*) = \mathcal{D}(\lambda)$ ，因此可得 $f(x^*, z^*) + g(y^*, z^*) \geq p^*$ 。另一方面，如(2.9)中之定義，由於 $p^* = \max_{x,y,z} f(x, z) + g(y, z)$ ，因此 $p^* \geq f(x^*, z^*) + g(y^*, z^*)$ 。由上可知， $p^* = f(x^*, z^*) + g(y^*, z^*)$ ；換句話說，對偶分解法中，原始問題之最佳目標值，一定會與對偶問題之最佳目標值相同。

2.5 資源分配與公平性[15]

在本篇論文中，我們考慮客戶與AP之間的連線問題。我們將引入效用函數(utility function)，並整理成最佳化問題進行求解。為了了解效用函數如何設計，在此節介紹幾種眾所皆知的公平性原則，如下所述[15]：

- 具權值的比例公平性(Weighted proportional fairness)；
- 最小潛在延遲公平性(minimum potential delay fairness)；
- 最大最小公平性(max-min fairness)。

2.5.1 最佳化架構

令 $U_i(r_i)$ 是 r_i 的效用函數，其中 r_i 為用戶 i 之速率，並假設 $U_i(\cdot)$ 為平滑（連續可）的凹函數。效用函數應體現經濟學中收益遞減法則(law of diminishing returns)的概念；也就是說，儘管增加擁有的資源，可增加總價值。但持續擁有更多的資源，增長的貢獻會愈來愈少。自然對數函數就是這種行為的一個例子，如圖2-5所示。另一方面，資源分配應考慮網絡中鏈路的容量限制，假設所有鏈路 l 都具有固定容量。

資源分配的問題可表示為(2.17)：

$$\max \quad \sum_{\{r_i\} \in \mathcal{S}} U_i(r_i) \quad (2.17a)$$

$$\text{Subject to } \sum_{\{r_i\} \in \mathcal{S}} r_i \leq c_l, l \in \mathcal{L}, \quad (2.17b)$$

$$r_i \geq 0, i \in \mathcal{C}. \quad (2.17c)$$

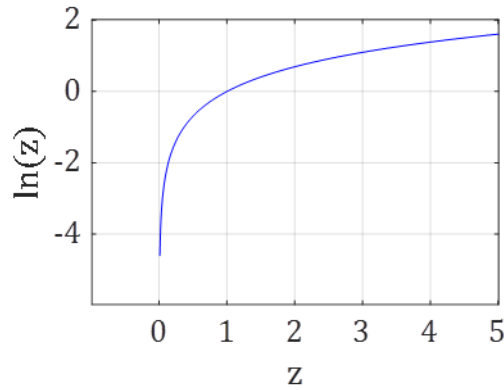


圖2-5:自然對數的函數圖像

其中， C 為用戶 i 的集合。 \mathcal{L} 為鏈路 l 的集合。 c_l 為鏈路 l 的容量。

2.5.2 公平性原則

接著，我們介紹一些公平性原則、其所對應之效用函數、以及其特點。

A. 具權值的比例公平性

根據比例公平性(proportional fairness)原則，可得 $U_i(r_i) = \log r_i$ 。若加上權重 w_i ，則成為具權值的比例公平性，或 $U_i(r_i) = w_i \log r_i$ 。稍後，在3.1節中，我們將考慮此公平性原則，並將有更進一步的介紹與討論。

B. 最小潛在延遲公平性

若考慮最小潛在延遲公平性原則，則 $U_i(r_i) = -\frac{w_i}{r_i}$ ，其理由如下。假設用戶 i 正嘗試傳送大小為 w_i 的資料。由於用戶 i 的速率為 r_i ， $\frac{w_i}{r_i}$ 為傳輸所需花費的時間。除此之外，由於花費時間要愈少愈好，但(2.17a)為最大化，效用函數因此乘上負號。所以，效用函數表示為 $U_i(r_i) = -\frac{w_i}{r_i}$ 。

C. 最大最小公平性

一般來說，我們知道最大最小公平性的目標為將最小的分配最大化。也就是說，它試圖為獲得最小資源的客戶提供最大的保護。因此，若資源分配已達到最大最小公平性原則，則在不減少獲得較小的人的分配的情況下，任何人的分配都無法進一步增加。換句話說，當已達到最大最小公平性時，如果不讓資源稀少的人變得更窮，就無法更進一步增加任何人的分配。

D. 通用的公平性 α -fairness

[15] 定義一個通用的效用函數，可同時涵蓋上述不同的公平性原則。甚至，藉由改變參數值 α ，可涵蓋其他的公平性原則：

$$U_i(r_i) = w_i \frac{r_i^{1-\alpha}}{1-\alpha}, \quad \alpha > 0, \alpha \neq 1. \quad (2.18)$$

可發現當 $\alpha \rightarrow 1$ 時， $U_i(r_i) = w_i \log r_i$ ，可對應具權值的比例公平性。當 $\alpha \rightarrow 2$ 時， $U_i(r_i) = -\frac{w_i}{r_i}$ ，可對應最小潛在延遲公平性。當 $\alpha \rightarrow \infty$ 時，可對應最大最小公平性。

根據以上的描述，(2.18) 可以改寫為：

$$U_i(r_i) = \begin{cases} w_i \frac{r_i^{1-\alpha}}{1-\alpha}, & \alpha > 0, \alpha \neq 1 \\ w_i \log r_i, & \alpha = 1 \end{cases} \quad (2.19)$$

第三章 分析方法

在本章中，我們根據比例公平性(proportional fairness)的原則，針對不同的目標(objective)與限制(contraint)，探討如何設計對應的演算法。首先，考慮用戶傳輸需求為已知時，集中式(centralized)演算法之設計。接著，我們討論當目標函數加入權重(weight)時，分散式(distributed)的演算法設計。

3.1 連線最佳化

在本節中，將討論5G毫米波網路中，用戶(client)連線的最佳化問題(association optimization)，其中，包含用戶的連線(association)與接入點(access point, AP)的資源分配(resource allocation)。當考慮該連線最佳化問題時，我們引入效用函數(utility function)。由於效用函數一般而言應綜合考慮如負載平衡(load balance)、公平性(fairness)、系統吞吐量(throughput)等。因此，用戶連線的最佳化，可等同於效用函數之最大化(maximization)。

接下來，我們分別以 \mathcal{C} 代表所有用戶 i 的集合， \mathcal{A} 代表所有AP k 的集合。須注意在本論文中，我們僅考慮每個用戶只允許與一個AP進行連線。於是，連線最佳化問題可表示為：

$$\max_{\mathbf{x}, \mathbf{r}, \mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} \mathcal{U}(r_{ik} y_{ik}) \quad (3.1a)$$

$$\text{s. t. } \sum_{k \in \mathcal{A}} x_{ik} = 1, \quad x_{ik} \in \{0, 1\}, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \quad (3.1b)$$

$$\sum_{i \in \mathcal{C}} y_{ik} \leq 1 \quad y_{ik} \geq 0, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \quad (3.1c)$$

其中， $\mathbf{x} = \{x_{ik} | i \in \mathcal{C}, k \in \mathcal{A}\}$ ， $\mathbf{r} = \{r_{ik} | i \in \mathcal{C}, k \in \mathcal{A}\}$ ， $\mathbf{y} = \{y_{ik} | i \in \mathcal{C}, k \in \mathcal{A}\}$ 。(3.1)的詳細說明如後。

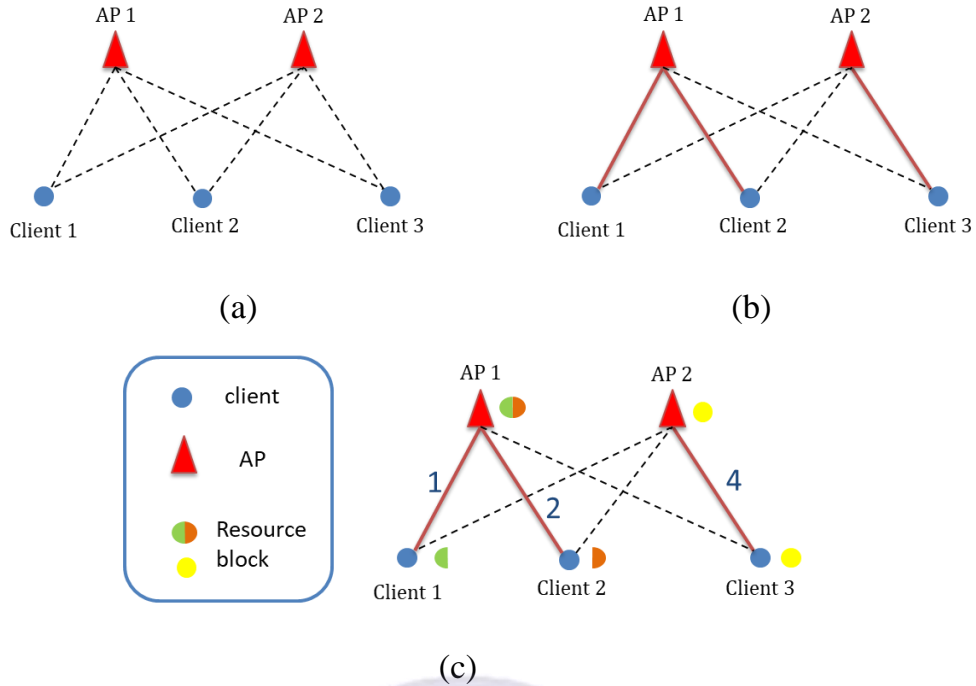


圖 3-1:用戶與 AP 之間的連線與資源分配範例；(a)所有連線可能性；(b)連線結果；(c)各個AP的資源分配。

首先，(3.1a)與(3.1b)中， x_{ik} 為二元連線指示器 (binary association indicators)；若用戶 i 與AP k 存在連線， $x_{ik} = 1$ ；否則， $x_{ik} = 0$ 。以圖3-1 為例，假設 3 個用戶，2 個AP。若每個用戶與任何一個AP皆可以進行連線，因此有 6 種連線可能，分別以虛線表示，如圖3-1(a)中虛線。但是，如前所述，由於每個用戶只允許與其中一個AP進行連線，連線結果可因此如圖3-1(b)中紅色實線。該結果可表示為 $x_{11} = x_{21} = x_{32} = 1$ ，且 $x_{12} = x_{22} = x_{31} = 0$ 。明顯地，它是其中一組可行解(feasible solution)。

其次，在(3.1a)中， r_{ik} 為用戶 i 與 AP k 之間的連線速率，該數值與兩者之間的通道狀況相關；因此，若連線結果確定，該數值可唯一的被決定。此外， y_{ik} 為AP k 分配給用戶 i 的資源，也就是所分配時間之比例；仍以圖3-1為

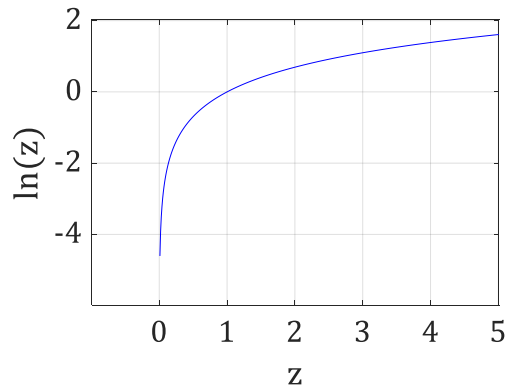


圖3-2:自然對數函數。

例，圖3-1(c)為資源分配之結果。其中， $y_{11} = \frac{1}{2}$ ， $y_{21} = \frac{1}{2}$ ， $y_{22} = 1$ 。須注意 AP 1並非在同一個瞬間與 2 個用戶同時進行連線；舉例來說， $y_{11} = \frac{1}{2}$ 意味著用戶1與 AP 1連線的時間，占AP 1全部的二分之一；當兩者連線期間，另一個用戶(也就是用戶2)為閒置。另一方面，紅線旁邊的數字則表示用戶與AP連線時的速率；如圖3-1(c)所示， $r_{11} = 1$ ， $r_{21} = 2$ ， $r_{22} = 4$ 。另外，(3.1b)限制每個用戶*i*只能與一個AP *k*相連線；而(3.1c)限制一個AP的資源分配，其總和不能超過 1，如圖 3-1(c)所示。

最後，(3.1a)中， $U(r_{ik} \cdot y_{ik})$ 為效用函數(utility function)。在 2.5 節中提到，效用函數應考慮負載平衡、公平性、吞吐量等。除此之外，2.5 節中也提到效用函數應體現經濟學中收益遞減法則(law of diminishing returns)的概念[15]；也就是說，儘管增加擁有的資源可增加總價值，但持續擁有更多的資源，增長的貢獻會愈來愈少。根據上面的說明，效用函數可如圖 3-2 之對數函數。所以，在(3.1a)中，效用函數將以自然對數函數來取代。該自然對數函數之效用函數，一般被稱為比例公平性，可使得 AP 間達到負載平衡。關於這個部分，稍後(3.9)會進一步解釋。於是，在比例公平性的原則之下，(3.1)可改寫為：

$$\begin{aligned}
& \max_{\mathbf{x}, \mathbf{r}, \mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} \log r_{ik} y_{ik} \\
& \text{s. t. } \sum_{k \in \mathcal{A}} x_{ik} = 1, \quad x_{ik} \in \{0, 1\}, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \\
& \quad \quad \sum_{i \in \mathcal{C}} y_{ik} \leq 1, \quad y_{ik} \geq 0, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A},
\end{aligned} \tag{3.2}$$

其中， $\mathbf{x}, \mathbf{r}, \mathbf{y}$ 分別代表 x_{ik}, r_{ik}, y_{ik} 所構成之向量。

3.2 集中式演算法

在本節中，我們參考[6]的做法，將(3.2)之連線最佳化問題改寫為一個多維度 \mathbf{x} 的配置問題。然後，利用對偶分解(dual decomposition)的方法，以獲得最佳的目標值。根據小型範例的測試結果顯示，該目標值與窮舉法的最佳目標值相同；然而，由於求解過程不同，本論文之方法可以更快速找到最佳解。

考慮(3.2)並假設 x_{ik} 是已知的，也就是假設 \mathbf{x} 是一個可行解。然後，便可藉此求得(3.2)對應之 r_{ik} 與最佳之 y_{ik} ，其過程介紹如後。首先，可將(3.2)之目標改寫為：

$$\begin{aligned}
\max_{\mathbf{x}, \mathbf{r}, \mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} \log r_{ik} y_{ik} &= \max_{\mathbf{x}, \mathbf{r}, \mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} (\log r_{ik} + \log y_{ik}) \\
&= \max_{\mathbf{x}, \mathbf{r}, \mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} (x_{ik} \log r_{ik} + x_{ik} \log y_{ik}).
\end{aligned} \tag{3.3}$$

根據香濃定理(Shannon's theorem)， r_{ik} 可以根據下式求得：

$$r_{ik} = B \log_2(1 + SNR_{ik}), \tag{3.4}$$

其中 B 為頻寬， SNR_{ik} 為用戶 i 與 AP k 之間鏈路的訊號雜訊功率比 (signal-to-noise ratio, SNR)，或簡稱訊雜比。該數值可以透過測量用戶 i 與 AP k 之間的距離後計算得知，詳細計算如 2.2 節所述。

令 n_k 為 AP k 的用戶連線數量，也就是：

$$n_k = \sum_{i \in \mathcal{C}} x_{ik}. \quad (3.5)$$

根據(3.3)且假設 x_{ik} 為已知，計算最佳的 y_{ik} 相當於下列最佳化問題，其中 $y_{ik} > 0$ ，且 $\sum_{i \in \mathcal{C}} y_{ik} \leq 1$ ：

$$\max_{\mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} \log y_{ik} = \max_{\mathbf{y}} \sum_{k \in \mathcal{A}} \sum_{i \in \mathcal{C}_k} \log y_{ik}. \quad (3.6)$$

其中 \mathcal{C}_k 代表與 AP k 連線用戶的集合，而該集合中用戶的數量應為 n_k 。透過簡單的對數運算，(3.6) 可以輕易地改寫為：

$$\max_{\mathbf{y}} \sum_{k \in \mathcal{A}} \sum_{i \in \mathcal{C}_k} \log y_{ik} = \max_{\mathbf{y}} \sum_{k \in \mathcal{A}} \log \prod_{i=1}^{n_k} y_{ik}. \quad (3.7)$$

另外，由於對數為遞增函數，(3.7) 等同於下列最佳化問題：

$$\begin{aligned} \max_{\mathbf{y} > \mathbf{0}} \quad & \prod_{i=1}^{n_k} y_{ik}, \\ \text{s. t.} \quad & \sum_{i=1}^{n_k} y_{ik} \leq 1. \end{aligned} \quad (3.8)$$

然後，根據算幾不等式 (arithmetic-geometric mean inequality)，可以得知當 $y_{1k} = y_{2k} = \dots = y_{n_k} = 1/n_k$ 時，可以最大化目標值，關於算幾不等式之說

明與證明，詳見附錄A。於是，我們可以求得最佳之 y_{ik} 為：

$$y_{ik} = \frac{1}{n_k}. \quad (3.9)$$

所以，(3.2)之目標可以改寫為：

$$\max_{x,r,n} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} \log r_{ik} - \sum_{k \in \mathcal{A}} n_k \log n_k. \quad (3.10)$$

令 $b_{ik} = \log r_{ik}$ ，可以使得(3.2)成為：

$$\begin{aligned} \max_{x,r,y} \quad & \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} b_{ik} - \sum_{k \in \mathcal{A}} n_k \log n_k \\ \text{s. t.} \quad & \sum_{k \in \mathcal{A}} x_{ik} = 1, \quad x_{ik} \in \{0,1\}, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \\ & \sum_{i \in \mathcal{C}} y_{ik} \leq 1, \quad y_{ik} \geq 0, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \end{aligned} \quad (3.11)$$

接下來，我們利用拉格朗日乘數法(Lagrangian multiplier)，進一步處理(3.5)之限制與(3.11)。為此，我們必須放寬 x_{ik} ，以使得其數值可為介於0與1間之任何實數，而非原本0或1之整數。令 $\lambda = \{\lambda_k | k \in \mathcal{A}\}$ ，可得拉格朗日函數(Lagrangian function)如下：

$$\begin{aligned} \mathcal{L}(x, n, \lambda) &= \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} b_{ik} - \sum_{k \in \mathcal{A}} n_k \log n_k + \sum_{k \in \mathcal{A}} \lambda_k (n_k - \sum_{i \in \mathcal{C}} x_{ik}) \\ &= \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} (b_{ik} - \lambda_k) + \sum_{k \in \mathcal{A}} n_k (\lambda_k - \log n_k). \end{aligned} \quad (3.12)$$

然後，可利用如 2.4 節所介紹之對偶分解(dual decomposition)方法，將(3.11)所對應之對偶方程式(Lagrangian dual function)寫成(3.13)：

$$\min_{\lambda} \mathcal{L}(\mathbf{x}, \mathbf{n}, \lambda) = \min_{\lambda} f(\mathbf{n}, \lambda) + g(\mathbf{x}, \lambda) \quad (3.13a)$$

其中，

$$f(\mathbf{n}, \lambda) = \max_{\mathbf{n} \geq 0} \sum_{k \in \mathcal{A}} n_k (\lambda_k - \log n_k) \quad (3.13b)$$

$$g(\mathbf{x}, \lambda) = \begin{cases} \max_{\mathbf{x}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} (b_{ik} - \lambda_k) x_{ik} \\ \text{s.t. } x_{ik} \in [0, 1], \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \end{cases} \quad (3.13c)$$

(3.1b)

關於(3.13)如何求解，詳述於後。根據 2.4 節對偶分解法之介紹，(3.13b) 與(3.13c)會分別根據目前的 λ ，計算對應最佳的 \mathbf{n} 與 \mathbf{x} 。然後，針對該求得之 \mathbf{n} 與 \mathbf{x} ，(3.13a)會判斷是否已經來到最佳解。如果不是最佳解，會繼續產生新的 λ 。

- 首先，關於(3.13c)，其最大化過程如下。對於每個用戶 i ，各個維度的斜率為：

$$\frac{\partial g}{\partial x_{ik}} = (b_{ik} - \lambda_k), \quad (3.14)$$

由於，根據(3.1b)， $\sum_{k \in \mathcal{A}} x_{ik} = 1$ ；因此，每個用戶 i 都必須利用有限的數值 x_{ik} ，以達到最大化。為此，用戶 i 將選擇具有最大斜率 $(b_{ik} - \lambda_k)$ 的維度 k ，讓 $x_{ik} = 1$ ，其餘為0。如此，可以得到(3.13c)最佳之 \mathbf{x} 。

- 接著，關於(3.13b)，可直接對 n_k 偏微分，以獲得該方向之斜率。然後，使斜率為0，以獲得最大化之 n_k 。也就是

$$\frac{\partial f}{\partial n_k} = \lambda_k - \frac{\partial}{\partial n_k} (n_k \log n_k) = \lambda_k - (\log n_k + 1) = 0, \quad (3.15)$$

根據(3.15)，可得 $n_k = e^{(\lambda_k - 1)}$ 。收集所有 n_k ，可得(3.13b)最佳之 \mathbf{n} 。

- 最後，對於(3.13a)而言，其斜率若為0時，則來到最佳解。也就是

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \lambda_k} &= \frac{\partial}{\partial \lambda_k} (\sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i (b_{ik} - \lambda_k) + \sum_{k \in \mathcal{A}} n_k (\lambda_k - \log n_k)) \\ &= n_k - \sum_{i \in \mathcal{C}} x_{ik}.\end{aligned}\quad (3.16)$$

因此，若對於所有的 k ， $n_k - \sum_{i \in \mathcal{C}} x_{ik}$ 全為0，(3.13)已求得最佳解 \mathbf{x} ；否則， λ_k 將變換新的數值，其變換幅度與各個維度的斜率 $n_k - \sum_{i \in \mathcal{C}} x_{ik}$ 成正比。

最後，我們用稍後圖3-3之演算法1計算(3.13)的最佳解。須注意，(3.2)-(3.3)、(3.6)-(3.7)、(3.10)-(3.12)、(3.13b)之對數，皆為自然對數。另外，由(3.13c)可以發現儘管 x_{ik} 放寬在0到1間之任何實數，但是當得到最佳解時，其所對應之 x_{ik} 會非0即1。其理由如下。首先，在某次遞迴中，由於 b_{ik} 與 λ_k 為定值， $(b_{ik} - \lambda_k)$ 因此為已知。根據(3.1b)， $\sum_{k \in \mathcal{A}} x_{ik} = 1$ 。於是，如要最大化(3.13c)之目標，用戶 i 會選擇所有 $(b_{ik} - \lambda_k)$ 當中，數值最大所對應之AP k ，也就是 $k^* = \arg \max_k (b_{ik} - \lambda_k)$ ；於是， $x_{ik^*} = 1$ ； $x_{ik} = 0$ ， $\forall k \neq k^*$ 。因此，該放寬並不影響 x_{ik} 必須非0即1之限制。

須注意隨著放寬 x_{ik} 使其數值為介於0與1間之任何實數，(3.13a)為(3.11)的對偶問題(dual problem)。由於(3.11)與(3.5)皆為線性等式(linear equalities)；而且，明顯地存在可行解。以圖3-1為例，圖3-1(b)或3-1(c)即為其中一個可行解。根據斯萊特的條件(Slater condition)，強對偶性(strong duality)將成立，詳見附錄B。換句話說，(3.11)與(3.13a)所獲得之最佳目標值會相同。

```

1: //input  $\mathcal{A}, \mathcal{C}, b_{ik}, \lambda_k$ 
2: Given  $\mathcal{A}$  ;
3: Given  $\mathcal{C}$  ;
4: Given  $b_{ik}, \forall i \in \mathcal{C}, \forall k \in \mathcal{A}$ ;
5:  $x_{ik} = 0, \forall i \in \mathcal{C}, \forall k \in \mathcal{A}$ ;
6:  $n_k = 0, \forall k \in \mathcal{A}$ ;
7:  $\lambda_k = 0, \forall k \in \mathcal{A}$ ;
8:  $\delta = 0.01$ ; //內建基本步伐
9:
10: //code:
11: do
12:      $n_k = e^{(\lambda_k - 1)}$  ;
13:      $x_{ik} = \arg \max_x \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} (b_{ik} - \lambda_k) x_{ik}$ ;
14:      $\lambda_k = \lambda_k - \delta \times (n_k - \sum_{i \in \mathcal{C}} x_{ik})$ ;
15: while  $0 < x_{ik} < 1$  AND  $\sum_{k \in \mathcal{A}} x_{ik} = 1$  AND  $n_k - \sum_{i \in \mathcal{C}} x_{ik} \neq 0$ 
16: end while
17: return  $\mathbf{x}$ .
```

圖 3-3:演算法 1。

關於演算法1的說明如下。首先，設定每個 λ_k 之初始值。由於每個 b_{ik} 為已知，11-13行可計算出對應 λ 之最佳 \mathbf{n} 與 \mathbf{x} ；接著，在第15行判定演算法是否終止。該終止的條件取決於(3.13a)是否已達到最佳，也就是每個維度的斜率 $\frac{\partial \mathcal{L}}{\partial \lambda_k} = (n_k - \sum_{i \in \mathcal{C}} x_{ik})$ 是否全為0。若不為0，第14行計算新的步伐幅度，會等於基礎幅度 δ 乘上各個維度的斜率，也就是 $(n_k - \sum_{i \in \mathcal{C}} x_{ik})$ 。然後，繼續進行11-15行之間的循環計算，直到終止條件成立為止。最後，輸出最佳解 \mathbf{x} 。須注意，在(3.13b)與(3.13c)中，由於 \mathbf{x} 、 \mathbf{n} 在演算法執行中皆為整數之向量。因此，演算法終止的條件，也就是每個 $(n_k - \sum_{i \in \mathcal{C}} x_{ik})$ 必須精確為0，而非僅只是接近0。

接下來，我們透過圖3-4之小型範例，以說明對偶分解(dual decomposition)方法或演算法1的計算過程。與圖3-1之範例類似，仍假設系統中有2個AP，3個用戶；而且，用戶 i 與AP k 之 b_{ik} 如圖3-4(a)所示。根據窮舉法，最佳解之連線為 $x_{11} = x_{21} = x_{32} = 1$ ，其餘為0，如圖3-4(b)所示。

另一方面，對偶分解(dual decomposition)方法或演算法1，其過程如圖3-5所示。假設基本步伐 $\delta = 0.5$ ，且初始值 $\lambda_1 = 5$ 、 $\lambda_2 = 0$ 。須注意其中 $0 \times \log 0 \cong 0^1$ 。

¹ $0 \times \log 0 \cong 0$: 證明 $\lim_{x \rightarrow 0} x \log x = 0$ ，使用羅畢達法則(L'Hôpital's rule):

$$\lim_{x \rightarrow 0} x \log x = \lim_{x \rightarrow 0} \frac{\log x}{x^{-1}} = \lim_{x \rightarrow 0} \frac{x^{-1}}{-x^{-2}} = \lim_{x \rightarrow 0} -x = 0.$$

所以可以得出: $0 \times \log 0 = 0$ 。

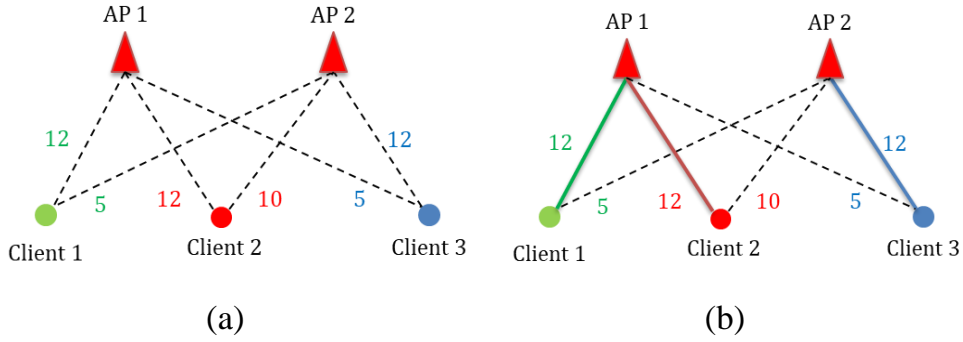


圖 3-4: (a)小型測試範例；(b)窮舉法最佳解之連線。

首先，第一輪中， $(\lambda_1, \lambda_2) = (5, 0)$ 。由於 (n_1, n_2) 有4種可能，分別為 $(3, 0)$ 、 $(2, 1)$ 、 $(1, 2)$ 、 $(0, 3)$ 。根據(3.13b)，

- $(n_1, n_2) = (3, 0)$ 對應之目標值為 $3(5 - \log 3) + 0(0 - \log 0) = 11.7$ ；
- $(n_1, n_2) = (2, 1)$ 對應之目標值為 $2(5 - \log 2) + 1(0 - \log 1) = 8.6$ ；
- $(n_1, n_2) = (1, 2)$ 對應之目標值為 $1(5 - \log 1) + 2(0 - \log 2) = 6.4$ ；
- $(n_1, n_2) = (0, 3)$ 對應之目標值為 $0(5 - \log 0) + 3(0 - \log 3) = -3.3$ 。

明顯地，由於其中最大目標值為11.7，對應的最佳解為 $(n_1, n_2) = (3, 0)$ 。其次，根據(3.13c)，用戶*i*將計算所有 $(b_{ik} - \lambda_k)$ ，然後與其中最大數值所對應之AP *k*進行連線。

- 對於用戶1來說，若與AP1連線，該數值為 $(12 - 5) = 7$ ；若與AP2連線，該數值為 $(5 - 0) = 5$ 。因此，用戶1會選擇與AP1進行連線。
- 對於用戶2來說，若與AP1連線，該數值為 $(12 - 5) = 7$ ；若與AP2連線，該數值為 $(10 - 0) = 10$ 。因此，用戶2會選擇與AP2進行連線。
- 對於用戶3來說，若與AP1連線，該數值為 $(5 - 5) = 0$ ；若與AP2連線，該數值為 $(12 - 0) = 12$ 。因此，用戶3會選擇與AP2進行連線。

因此，第一輪之用戶連線結果為 $x_{11} = x_{22} = x_{32} = 1$ ，其餘為0，如圖3-5(a)所示。最後，如前所述，我們檢查終止條件是否成立。也就是說，對於所有的*k*， $(n_k - \sum_{i \in C} x_{ik})$ 是否全為0。然而，

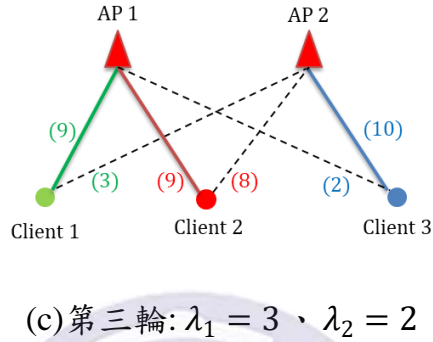
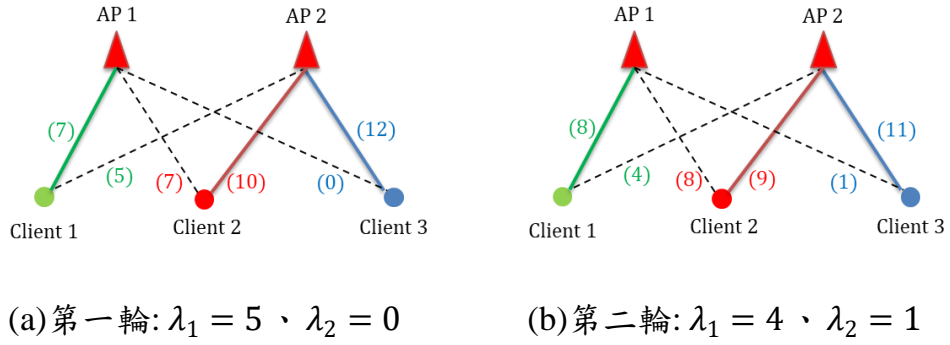


圖 3-5: 小型測試範例在演算法1之運算過程。

- 由於 $n_1 = 3$ ，但 $\sum_{i \in \mathcal{C}} x_{i1} = x_{11} + x_{21} + x_{31} = 1$ ；
- 且 $n_2 = 0$ ，但 $\sum_{i \in \mathcal{C}} x_{i2} = x_{12} + x_{22} + x_{32} = 2$ 。

因此，終止條件並未成立，必須繼續進行下一輪。而下一輪之 λ ，根據演算法1的17行，可計算為 $\lambda_1 = 5 - 0.5 \times (3 - 1) = 4$ 、與 $\lambda_2 = 0 - 0.5 \times (0 - 2) = 1$ ，或 $(\lambda_1, \lambda_2) = (4, 1)$ 。

接著，第二輪中， $(\lambda_1, \lambda_2) = (4, 1)$ 。由於 (n_1, n_2) 有4種可能，分別為 $(3, 0)$ 、 $(2, 1)$ 、 $(1, 2)$ 、 $(0, 3)$ ，根據(3.13b)，

- $(n_1, n_2) = (3, 0)$ 對應之目標值為 $3(4 - \log 3) + 0(1 - \log 0) = 8.7$ ；
- $(n_1, n_2) = (2, 1)$ 對應之目標值為 $2(4 - \log 2) + 1(1 - \log 1) = 7.6$ ；
- $(n_1, n_2) = (1, 2)$ 對應之目標值為 $1(4 - \log 1) + 2(1 - \log 2) = 4.6$ ；
- $(n_1, n_2) = (0, 3)$ 對應之目標值為 $0(4 - \log 0) + 3(1 - \log 3) = -0.3$ 。

明顯地，由於其中最大目標值為8.7，對應的最佳解為 $(n_1, n_2) = (3, 0)$ 。其次，

根據(3.13c)，用戶*i*將計算所有 $(b_{ik} - \lambda_k)$ ，然後與其中最大數值所對應之AP *k* 進行連線。

- 對於用戶1來說，若與AP1連線，該數值為 $(12 - 4) = 8$ ；若與AP2連線，該數值為 $(5 - 1) = 4$ 。因此，用戶1會選擇與AP1進行連線。
- 對於用戶2來說，若與AP1連線，該數值為 $(12 - 4) = 8$ ；若與AP2連線，該數值為 $(10 - 1) = 9$ 。因此，用戶2會選擇與AP2進行連線。
- 對於用戶3來說，若與AP1連線，該數值為 $(5 - 4) = 1$ ；若與AP2連線，該數值為 $(12 - 1) = 11$ 。因此，用戶3會選擇與AP2進行連線。

因此，第二輪之用戶連線結果為 $x_{11} = x_{22} = x_{32} = 1$ ，其餘為0，如圖3-5(b)所示。最後，如前所述，我們檢查終止條件是否成立。也就是說，對於所有的*k*， $(n_k - \sum_{i \in \mathcal{C}} x_{ik})$ 是否全為0。然而，

- 由於 $n_1 = 3$ ，但 $\sum_{i \in \mathcal{C}} x_{i1} = x_{11} + x_{21} + x_{31} = 1$ ；
- 且 $n_2 = 0$ ，但 $\sum_{i \in \mathcal{C}} x_{i2} = x_{12} + x_{22} + x_{32} = 2$ 。

因此，終止條件並未成立，必須繼續進行下一輪。而下一輪之 λ ，可計算為 $\lambda_1 = 4 - 0.5 \times (3 - 1) = 3$ 、與 $\lambda_2 = 1 - 0.5 \times (0 - 2) = 2$ 、或 $(\lambda_1, \lambda_2) = (3, 2)$ 。

最後，第三輪中， $(\lambda_1, \lambda_2) = (3, 2)$ 。其中 (n_1, n_2) 有4種可能，分別為 $(3, 0)$ 、 $(2, 1)$ 、 $(1, 2)$ 、 $(0, 3)$ ，根據(3.13b)，

- $(n_1, n_2) = (3, 0)$ 對應之目標值為 $3(3 - \log 3) + 0(2 - \log 0) = 5.7$ ；
- $(n_1, n_2) = (2, 1)$ 對應之目標值為 $2(3 - \log 2) + 1(2 - \log 1) = 6.6$ ；
- $(n_1, n_2) = (1, 2)$ 對應之目標值為 $1(3 - \log 1) + 2(2 - \log 2) = 5.6$ ；
- $(n_1, n_2) = (0, 3)$ 對應之目標值為 $0(3 - \log 0) + 3(2 - \log 3) = 2.7$ 。

明顯地，由於其中最大目標值為6.6，對應的最佳解為 $(n_1, n_2) = (2, 1)$ 。其次，根據(3.13c)，用戶*i*將計算所有 $(b_{ik} - \lambda_k)$ ，然後與其中最大數值所對應之AP *k*

進行連線。

- 對於用戶1來說，若與AP1連線，該數值為 $(12 - 3) = 9$ ；若與AP2連線，該數值為 $(5 - 2) = 3$ 。因此，用戶1會選擇與AP1進行連線。
- 對於用戶2來說，若與AP1連線，該數值為 $(12 - 3) = 9$ ；若與AP2連線，該數值為 $(10 - 2) = 8$ 。因此，用戶2會選擇與AP1進行連線。
- 對於用戶3來說，若與AP1連線，該數值為 $(5 - 3) = 2$ ；若與AP2連線，該數值為 $(12 - 2) = 10$ 。因此，用戶3會選擇與AP2進行連線。

因此，第三輪之用戶連線結果為 $x_{11} = x_{21} = x_{32} = 1$ ，其餘為0，如圖3-5(c)

所示。最後，我們檢查終止條件是否成立。也就是說，對於所有的 k ，

$(n_k - \sum_{i \in \mathcal{C}} x_{ik})$ 是否全為0:

- $n_1 = 2$ ，且 $\sum_{i \in \mathcal{C}} x_{i1} = x_{11} + x_{21} + x_{31} = 2$ ；
- $n_2 = 1$ ，且 $\sum_{i \in \mathcal{C}} x_{i2} = x_{12} + x_{22} + x_{32} = 1$ 。

因此，終止條件成立。因此，對偶分解(dual decomposition)方法之最佳解為 $x_{11} = x_{21} = x_{32} = 1$ ，其餘為0。

可以發現使用對偶分解法與窮舉法所得到的最佳解 x_{ik} 皆相同，分別如圖3-4(b)與3-5(c)所示。根據該最佳解 x_{ik} ，可透過(3.5)獲得 n_k 。然後，根據 n_k ，可利用(3.9)計算每個用戶之 y_{ik} 。於是，用戶連線與資源分配的問題，便可分別找到其最佳解 x 與 y 。須注意(3.13)不僅可以用演算法1的方式實現，也可以適用於分散的系統環境。換句話說，(3.13b)之最大化可以在AP端執行；而(3.13c)之最大化可以在用戶端執行。關於分散式的演算法設計，將於3.4節中詳細說明。

3.3 用戶頻寬的需求

關於資源分配，除了要顧及用戶間的公平性之外，也希望可以盡量滿足

每個用戶的需求(demand)。為此，我們在本節中加上用戶的需求限制，以使得 AP 能更針對性分配寶貴的通道資源。為此，我們增加用戶的需求限制如下：

$$y_{ik} \leq \frac{z_i}{r_{ik}}, \quad (3.17)$$

其中， z_i (bits/sec)代表用戶*i*所希望獲得的資料傳輸速率。如此一來，便可以避免AP分配過多資源給用戶*i*，AP也可將資源分配給其他有需要的用戶。

因此，(3.11)可以改寫為(3.18):

$$\max_{x,r,y} \quad \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} b_{ik} - \sum_{k \in \mathcal{A}} n_k \log n_k \quad (3.18a)$$

$$\text{s. t.} \quad \sum_{k \in \mathcal{A}} x_{ik} = 1, \quad x_{ik} \in \{0,1\}, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \quad (3.18b)$$

$$\sum_{i \in \mathcal{C}} y_{ik} \leq 1, \quad y_{ik} \geq 0, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \quad (3.18c)$$

$$y_{ik} \leq \frac{z_i}{r_{ik}}, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \quad (3.18d)$$

其中，由於(3.18a)、(3.18b)、(3.18c)與(3.11)完全相同，因此可使用3.2節的對偶分解方法，或演算法1，進行求解。但是，針對(3.18d)，我們會檢查演算法1之輸出。若其中包含已經滿足需求的用戶，除了將該用戶移至另一個集合，並將恰好滿足該用戶需求之資源，分配給這些用戶。然後，針對剩餘的用戶與剩餘的資源，重新執行演算法1，直到無法移動任何用戶到該集合為止。上述之完整作法如演算法2，如圖3-6所示，並詳細說明如下。

```

1: //input  $b_{ik}, \lambda_k, \mathcal{A}, \mathcal{C}$ 
2:  $\mathcal{N} = \{\emptyset\}$ ;
3: Given  $\mathcal{C}$ ;
4: Given  $\mathcal{A}$ ;
5: Given  $\mathcal{M}$ ; //包含所有連線的用戶
6: Given  $z_i, \forall i \in \mathcal{C}$ ;
7: Given  $b_{ik}, \forall i \in \mathcal{C}, \forall k \in \mathcal{A}$ ;
8:  $x_{ik} = 0, \forall i \in \mathcal{C}, \forall k \in \mathcal{A}$ ;
9:  $n_k = 0, \forall k \in \mathcal{A}$ ;
10:  $\lambda_k = 0, \forall k \in \mathcal{A}$ ;
11:  $y_{ik} = 0, \forall i \in \mathcal{C}, \forall k \in \mathcal{A}$ ;
12:  $m_k = 0, \forall k \in \mathcal{A}$ ;
13:
14: //code:
15: do
16:   Algorithm 1( $b_{ik}, \lambda_k, \mathcal{A}, \mathcal{M}$ );
17:    $y_{ik} = (1 - m_k) / n_k$ ;
18:   if  $y_{ik} > \frac{z_i}{r_{ik}}$  then
19:      $i \subseteq \mathcal{N}$ ;
20:      $\mathcal{M} = \mathcal{M} \setminus \{i\}$ ; //從集合中去掉滿足需求的用戶 i
21:   end if
22:
23:   if  $i \subseteq \mathcal{N}$  then
24:      $m_k = m_k + \frac{z_i}{r_{ik}}$ ; //已經用掉的資料量

```



```

25:     end if
26: while   $y_{ik} > \frac{z_i}{r_{ik}}$ 
27: end while
28: return  $\mathbf{x}$ .

```

圖 3-6:演算法 2。

首先，將所有用戶加入集合 \mathcal{M} 。每一輪，針對集合 \mathcal{M} 之用戶，在16、17行，會透過演算法1獲得連線解 \mathbf{x} 後，以計算出每個用戶新的資源分配 y_{ik} 。接著，在18-21行，檢查用戶 i 之 y_{ik} 是否已滿足其需求；若已滿足，則將該用戶移至集合 \mathcal{N} 。然後，在23-25行中，計算系統已使用之資源，或已配置給用戶之資源，會恰等於集合 \mathcal{N} 中所有用戶需求之總和。根據該計算結果，便可以在下一輪時，根據剩餘的資源決定新的資源分配 y_{ik} 。該演算法之終止條件，如26行，即集合 \mathcal{M} 是否仍存在新的資源分配足以滿足需求的用戶；若有，則繼續下一輪運算，直到集合 \mathcal{M} 中無法找到可滿足於新的資源分配之用戶為止。

須注意演算法2使得需求較小的用戶，在需求已滿足的前提下，將多餘的通道資源分配給需求尚未滿足的用戶，這是演算法1所無法做到的。關於演算法2之資源配置，可能發生下列3種情況：

- (a) 皆不滿足用戶之需求；
- (b) 皆滿足用戶之需求；
- (c) 部分滿足，但另一部分不滿足用戶之需求。

情況(a)發生於用戶所需要的通道資源皆大於AP所分配給用戶的通道資

源；情況(b)發生於用戶所需要的通道資源皆少於AP所分配的通道資源。上述兩種情況，使用演算法1即可完成資源配置。因此，演算法2主要對付(c)的情況。稍後，在第四章中，數值結果顯示當情況(c)發生時，使用演算法2，相較於演算法1，可以提高3倍之吞吐量。

3.4 用戶之權重

在連線的最佳化問題中，除了如3.3節用戶頻寬需求的限制之外，我們在本節考慮用戶之權重 w_i ，使得(3.2)改寫為(3.19):

$$\max_{x,r,y} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i \log r_{ik} y_{ik} \quad (3.19a)$$

$$\text{s. t. } \sum_{k \in \mathcal{A}} x_{ik} = 1, \quad x_{ik} \in \{0,1\}, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \quad (3.19b)$$

$$\sum_{i \in \mathcal{C}} y_{ik} \leq 1, \quad y_{ik} \geq 0, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \quad (3.19c)$$

類似3.2節之過程，假設 x_{ik} 是已知的，或 x 為可行解。然後，便可藉此求得該可行解對應之 r_{ik} 與最佳之 y_{ik} ，詳細過程說明如後。首先，類似3.2節之(3.3)，可將(3.19)之目標改寫為：

$$\begin{aligned} \max_{x,r,y} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i \log r_{ik} y_{ik} &= \max_{x,r,y} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i (\log r_{ik} + \log y_{ik}) \\ &= \max_{x,r,y} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} (x_{ik} w_i \log r_{ik} + x_{ik} w_i \log y_{ik}). \end{aligned} \quad (3.20)$$

其中， r_{ik} 可以從香濃定理(3.4) (Shannon's theorem)求得。因此，(3.20)前面部分為定值。為了計算 y_{ik} ，類似3.2節之(3.5)，令 n'_k 為所有與AP k連線的用戶權重之總和，也就是：

$$n'_k = \sum_{i \in \mathcal{C}} w_i x_{ik}. \quad (3.21)$$

由於 x_{ik} 是已知的，為了計算最佳之 y_{ik} ，類似3.2節之(3.6)，可將(3.20)之後面部分改寫為：

$$\max_{\mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i \log y_{ik} = \max_{\mathbf{y}} \sum_{k \in \mathcal{A}} \sum_{i \in \mathcal{C}_k} w_i \log y_{ik}. \quad (3.22)$$

其中， \mathcal{C}_k 代表所有與AP k連線用戶的集合。為了簡單說明起見，先假設權重 w_i 為整數，根據(3.21)，明顯地， n'_k 為整數。透過簡單的對數運算，類似3.2節之(3.7)，(3.22)可以改寫為：

$$\max_{\mathbf{y}} \sum_{k \in \mathcal{A}} \sum_{i \in \mathcal{C}_k} w_i \log y_{ik} = \max_{\mathbf{y}} \sum_{k \in \mathcal{A}} \log \prod_i^{n'_k} y_{ik}^{w_i}. \quad (3.23)$$

另外，類似3.2節之(3.8)，由於對數為遞增函數，(3.23)等效於下列最佳化問題：

$$\begin{aligned} \max_{\mathbf{y} > \mathbf{0}} \quad & \prod_i^{n'_k} y_{ik}^{w_i}, \\ \text{s. t.} \quad & \sum_{i=1}^{n'_k} y_{ik} \leq 1. \end{aligned} \quad (3.24)$$

首先，由於假設權重 w_i 為整數， $y_{ik}^{w_i}$ 可以看成是 w_i 個 y_{ik} 相乘，也就是把原本一個用戶i視為 w_i 個用戶i。所以，(3.24)可以視為AP k之總連線用戶數為 $\sum_i w_i$ 的資源分配。根據(3.18)，可以知道該總連線用戶數為 n'_k 。然後，類似3.2節之(3.9)，根據算幾不等式，詳見附錄A，可以得知當 $y_{1k} = y_{2k} = \dots = y_{n'_k k} = 1/n'_k$ 時，可以最大化(3.24)之目標值。最後，將 w_i 個用戶i重新合併視為1個用戶i。於是，我們可以求得最佳之 y_{ik} ，如下：

$$y_{ik} = \frac{w_i}{n'_k}. \quad (3.25)$$

須注意上述做法不僅只適用於 w_i 為整數，甚至它可以是有理數，詳細說明於附錄C。如3.2節之(3.11)，令 $b_{ik} = \log r_{ik}$ ，可以使得(3.19)之目標成為：

$$\begin{aligned}
 \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i \log r_i y_{ik} &= \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i \log r_i + \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i \log y_{ik} \\
 &= \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i b_{ik} + \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i \log \frac{1}{n'_k} \\
 &= \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i b_{ik} - \sum_{k \in \mathcal{A}} n'_k \log n'_k. \quad (3.26)
 \end{aligned}$$

也就是說，完整的(3.19)可寫成：

$$\begin{aligned}
 \max_{x, r, y} \quad & \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i b_{ik} - \sum_{k \in \mathcal{A}} n'_k \log n'_k \\
 \text{s. t.} \quad & \sum_{k \in \mathcal{A}} x_{ik} = 1, \quad x_{ik} \in \{0, 1\}, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \\
 & \sum_{i \in \mathcal{C}} y_{ik} \leq 1, \quad y_{ik} \geq 0, \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A},
 \end{aligned} \quad (3.27)$$

接下來，類似3.2節之(3.12)，我們利用拉格朗日乘數法，進一步處理(3.21)與(3.27)。為此，與3.2節相同，我們放寬 x_{ik} ，以使得其數值可為介於0與1間之任何實數，而非原本0或1之整數。令 $\lambda = \{\lambda_k | k \in \mathcal{A}\}$ ，可得拉格朗日函數如下：

$$\mathcal{L}(x, n', \lambda) = \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i (b_{ik} - \lambda_k) + \sum_{k \in \mathcal{A}} n'_k (\lambda_k - \log n'_k). \quad (3.28)$$

然後，類似3.2節之(3.13)，可利用如2.4節所介紹之對偶分解方法，將(3.27)所對應之對偶方程式寫成(3.29)：

$$\min_{\lambda} \mathcal{L}(\mathbf{x}, \mathbf{n}', \lambda) = \min_{\lambda} f(\mathbf{n}', \lambda) + g(\mathbf{x}, \lambda) \quad (3.29a)$$

其中，

$$f(\mathbf{n}', \lambda) = \max_{\mathbf{n}' \geq 0} \sum_{k \in \mathcal{A}} n'_k (\lambda_k - \log n'_k) \quad (3.29b)$$

$$g(\mathbf{x}, \lambda) = \begin{cases} \max_{\mathbf{x}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} w_i (b_{ik} - \lambda_k) x_{ik} \\ \text{s. t. } x_{ik} \in [0, 1], \quad \forall i \in \mathcal{C}, \quad k \in \mathcal{A}, \end{cases} \quad (3.29c)$$

(3.1b)

其中，由(3.29c)可以發現儘管 x_{ik} 放寬在0到1間之任何實數，但是當得到最佳解時，其所對應之 x_{ik} 會非0即1。其理由如 3.2 節(3.13c)之說明。因此，該放寬並不影響 x_{ik} 必須非0即1之限制。

關於(3.29)與(3.13)的異同，詳細討論如後。

- 首先，須注意(3.29)仍可採用(3.13)之演算法1求最佳解，唯其中第14行須改為 $\lambda_k = \lambda_k - \delta \times (n'_k - \sum_{i \in \mathcal{C}} w_i x_{ik})$ ；且第15行之終止條件變成判斷所有的 $(n'_k - \sum_{i \in \mathcal{C}} w_i x_{ik})$ 是否全為0。這是因為(3.29a)之斜率，類似於(3.16)，

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \lambda_k} &= \frac{\partial}{\partial \lambda_k} (\sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} x_{ik} w_i (b_{ik} - \lambda_k) + \sum_{k \in \mathcal{A}} n'_k (\lambda_k - \log n'_k)) \\ &= n'_k - \sum_{i \in \mathcal{C}} w_i x_{ik} . \end{aligned} \quad (3.30)$$

- 其次，對於(3.29b)，類似於(3.15)，可直接對 n'_k 偏微分，以獲得最大化之 n'_k ，即 $n'_k = e^{(\lambda_k - 1)}$ 。然而，須注意該 n'_k 與(3.15)之 n_k 不同；前者為用戶權重之總和，而後者為用戶數量。
- 接著，關於(3.29c)，類似於(3.14)，用戶 i 將選擇具有最大斜率 $w_i (b_{ik} - \lambda_k)$ 的維度 k ，讓 $x_{ik} = 1$ ，其餘為0。須注意對於用戶 i 來說，所有斜率 $w_i (b_{ik} - \lambda_k)$ 中的 w_i 值皆相同。這是因為權重 w_i 只因用戶而改變。因此，對用戶 i 來說， w_i 不影響最佳解 \mathbf{x} 的選擇，因此可被忽略。

演算法 3a (用戶端)

1. 測定用戶與AP之距離來推算 SNR，並從每個AP k接收 λ_k 廣播
 2. 解決 $x_{ik} = \arg \max_x \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{A}} w_i (b_{ik} - \lambda_k) x_{ik}$
-

演算法 3b (AP 端)

1. 解決(3.29b)藉由設定 $n'_k(t+1) = e^{(\lambda_k(t)-1)}$
 2. 更新 λ_k 藉由令 $\lambda_k(t+1) = \lambda_k(t) - \delta(t) \times (n'_k(t) - \sum_{i \in \mathcal{C}} w_i x_{ik})$
-

圖 3-7:演算法 3。

- 最後，(3.29)與(3.13)各自 λ_k 的演進變化並不同。在(3.13)中， λ_k 的演進變化會使得每個AP所連線的用戶數量趨向相同；然而，在(3.29)中， λ_k 的演進變化會使得每個AP所連線的用戶權重之總和趨向相同。如此一來，導致(3.13)與(3.29)之最佳解不相同。除此之外，(3.29)增加考慮用戶的權重，除了造成斜率改變為 $(n'_k - \sum_{i \in \mathcal{C}} w_i x_{ik})$ ，或步伐幅度變更為 $n'_k - \sum_{i \in \mathcal{C}} w_i x_{ik}$ ；並使得資源分配 $y_{ik} = \frac{w_i}{n'_k}$ 。須注意，根據(3.21)， $n'_k = \sum_{i \in \mathcal{C}_k} w_i$ ，使得 $y_{ik} = \frac{w_i}{\sum_{i \in \mathcal{C}_k} w_i}$ ；換句話說，AP會根據權重的比例進行分配。

(3.29)可以用分散式的方法求解。為此，我們提出演算法3，如圖3-7。根據演算法3，用戶端可以測定用戶i與AP k之距離，然後，依照2.2節所述計算SNR，並如3.2節的方式得到 b_{ik} 。然後，每個AP會傳送 λ_k 。用戶收到 λ_k 後，會求解(3.29c)，試圖決定其 x_{ik} ，或確定該用戶要與哪個AP進行連線。這些過程發生在用戶端，即演算法3a。然後，在AP端(演算法3b)，計算(3.29b)的最佳解，如前所述，也就是 $n'_k = e^{(\lambda_k-1)}$ ；並更新步伐 λ_k ，其變化幅度正比於 $n'_k - \sum_{i \in \mathcal{C}} w_i x_{ik}$ 。

第四章 數值結果與討論

在本章中，我們透過電腦模擬，以獲得第三章演算法之結果，並檢驗本論文所提出連線最佳化問題之性能，其中，模擬使用MATLAB軟體。首先，4.1節介紹實驗環境的設置。接著，4.2、4.3節分別呈現最佳化問題在演算法1、2的結果，並比較兩者之間的差別、優劣；除此之外，我們展現加入考慮用戶需求的演算法2，其吞吐量可顯著優於演算法1。最後，4.4節評估演算法3之表現，比較加入權重前後，最佳化問題之變化。

4.1 實驗環境

表 2 :模擬所採用之參數

參數名(Parameters)	參數值(Value)
頻寬	1 GHz [6]
頻段	28 GHz [6]
雜訊功率頻譜密度	-174 dBm/Hz [6]
接收端的噪音因數(noise figure, NF)	6 dB [6]
用戶數量	150 [6]
接入點(access point, AP)數量	5 [6]

在本節中，為了清楚地展現連線最佳化問題中，用戶的連線與AP的資源分配，我們設定較實際的參數與系統環境，以模擬第3章所提出之演算法，包含演算法1、2和3。MATLAB模擬所使用的相關參數，如表2所列。除此之外，如第三章所述， z_i 為用戶*i*所希望獲得的資料傳輸速率， w_i 為用戶*i*的權重，將成為考慮用戶需求時所使用的參數。

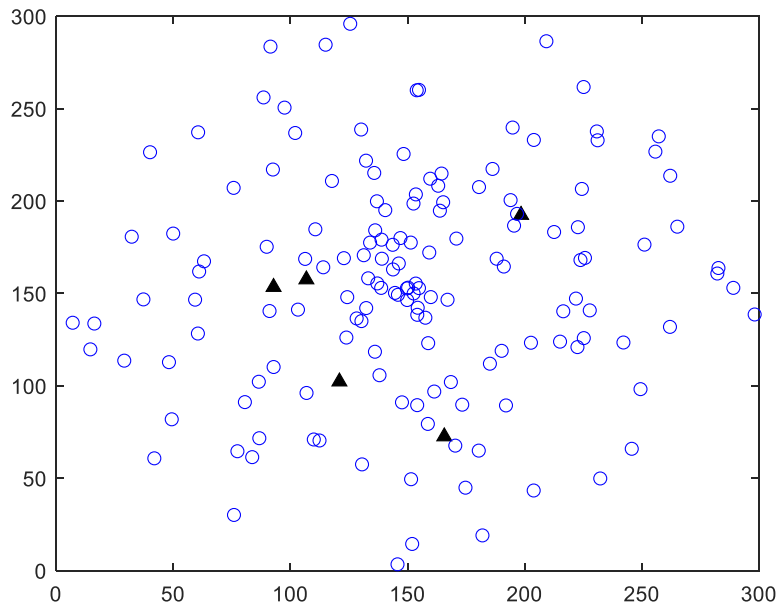


圖 4-1：AP 與用戶的位置分布結果圖。

MATLAB模擬的系統環境設定說明如後。首先，我們先獲得 AP 與用戶的位置，以取得兩者間的距離 d_{ik} 。為此，我們讓 AP 的位置，如圖4-1以(150,150)為中心、80 m為半徑，均勻分布於該範圍中。同樣地，用戶位置，仍以(150,150)為中心、150 m為半徑，均勻分布於該範圍中。位置分布結果，可如圖 4-1 所示，其中黑色三角形為AP的位置，藍色空心圓為用戶的位置。

4.2 連線最佳化問題的結果與討論

在決定AP與用戶的位置之後，即可使用MATLAB針對最佳化問題(3.2)，或演算法1進行模擬。其結果如圖4-2，顯示演算法1模擬後，用戶的連線選擇。

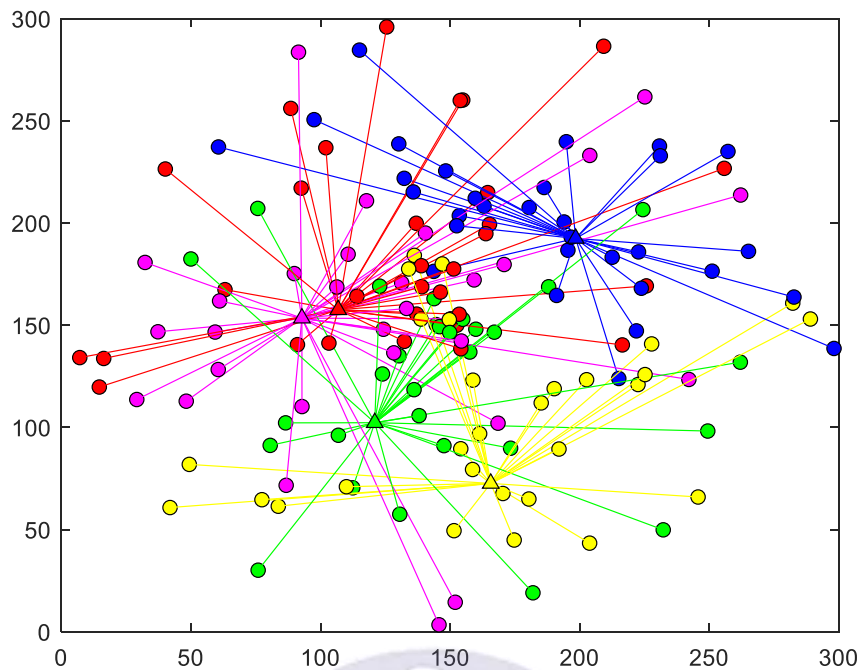


圖 4-2：AP 與用戶的連線結果圖。其中，不同的連線顏色代表用戶與對應顏色的 AP 進行連線。

4.2.1. 分析部分

如圖4-2，用戶的連線選擇說明如後。首先，根據3.2節之(3.13)，用戶會傾向於與較近的AP進行連線。這是因為，根據(3.13c)，各維度的斜率分別為 $(b_{ik} - \lambda_k)$ ，(3.13c)之最大化，會使得用戶選擇與具有最大斜率的AP進行連線。因此，距離愈小，訊雜比愈大，如 2.2 節之介紹。換言之， b_{ik} 愈大，斜率愈大。如圖4-2，可以看出AP旁的用戶，很多都與AP同顏色，說明大部分用戶選擇與較近的AP進行連線。

接著，可以發現並非所有用戶都與最近的AP進行連線，少部分用戶仍會選擇較遠、但負載較小的AP進行連線。其理由如後。

根據演算法1, 只有終止條件 $n_k - \sum_{i \in C} x_{ik} = 0$ 成立時, 才可來到最佳解。然而, 根據(3.13b), λ_k 會影響 n_k 的變化, 即 $n_k = e^{(\lambda_k - 1)}$; 也就是說, 若 λ_k 愈大, n_k 愈大。但是, 如前所述, 根據(3.13c), 各維度的斜率為 $(b_{ik} - \lambda_k)$, 若 λ_k 愈大, 將使得 $b_{ik} - \lambda_k$ 愈小, 與AP k連線的用戶愈少, 或 $\sum_{i \in C} x_{ik}$ 愈小。事實上, 整個會是一個權衡(trade-off)的過程。因此, 即便AP離某個用戶較遠, 若其負載 n_k 較小, λ_k 會因此較小, 使用戶與其連線的可能性增加。

最後, 根據實驗結果, 發現與每個AP連線的用戶數量趨向相同, 其理由如後。由於 b_{ik} 數值大多在22上下, 差異不大; 而根據(3.13c), 用戶選擇與具有最大斜率 $(b_{ik} - \lambda_k)$ 的AP進行連線。若 λ_k 間存在些微差距, 將導致許多用戶選擇其中最小 λ_k 之AP。根據上述之權衡過程, 將使得下一輪的該 λ_k 數值變大。最終, λ_k 間差距其實極小。於是, 根據(3.13b), $n_k = e^{(\lambda_k - 1)}$, λ_k 間差距極小使得每個AP的用戶數量幾乎相同。

4.2.2. 演進過程

這個部分, 將展示模擬過程中, 其相關參數的演進過程, 如圖4-3、圖4-4、圖4-5。首先, 我們觀察 λ_k 之演進變化, 如圖4-3。根據(3.13b), λ_k 會影響 n_k 的變化, 即 $n_k = e^{(\lambda_k - 1)}$ 。須注意, 根據實際模擬, 與每個AP連線的用戶數量會趨向相同, 也就是 $\frac{150}{5} = 30$, 使每個AP所連線之用戶數量約在30上下, 也就是 $n_k \approx 30$ 。因此, $\lambda_k \approx \ln 30 + 1 \approx 4.4$ 。圖4-3為 λ_k 之演進過程, 其中 λ_k 之初始值皆設定為4。

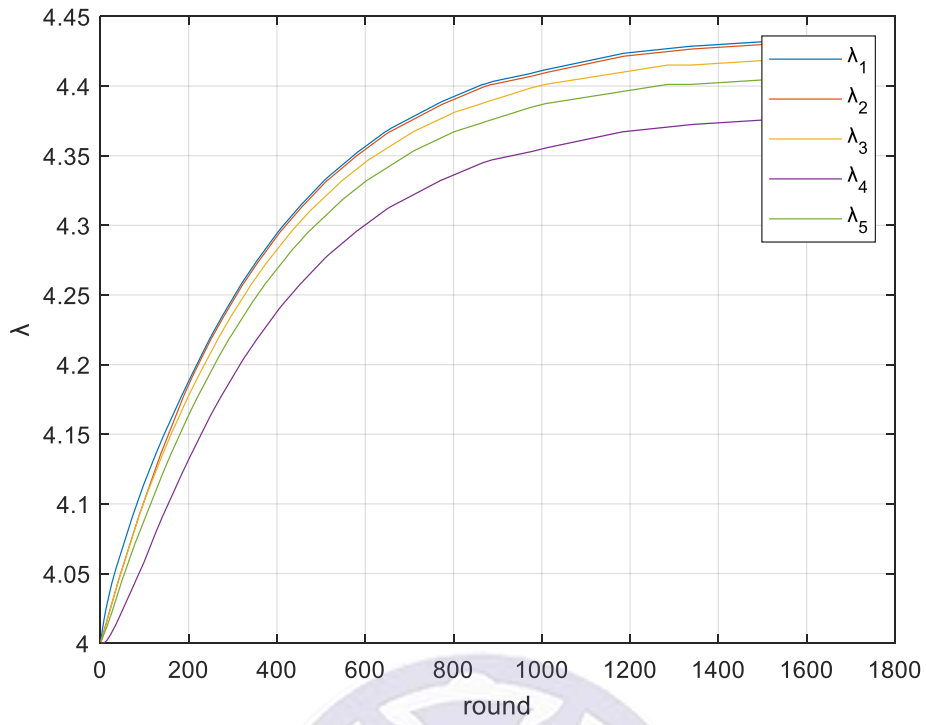


圖 4-3： λ_k 的演進過程。

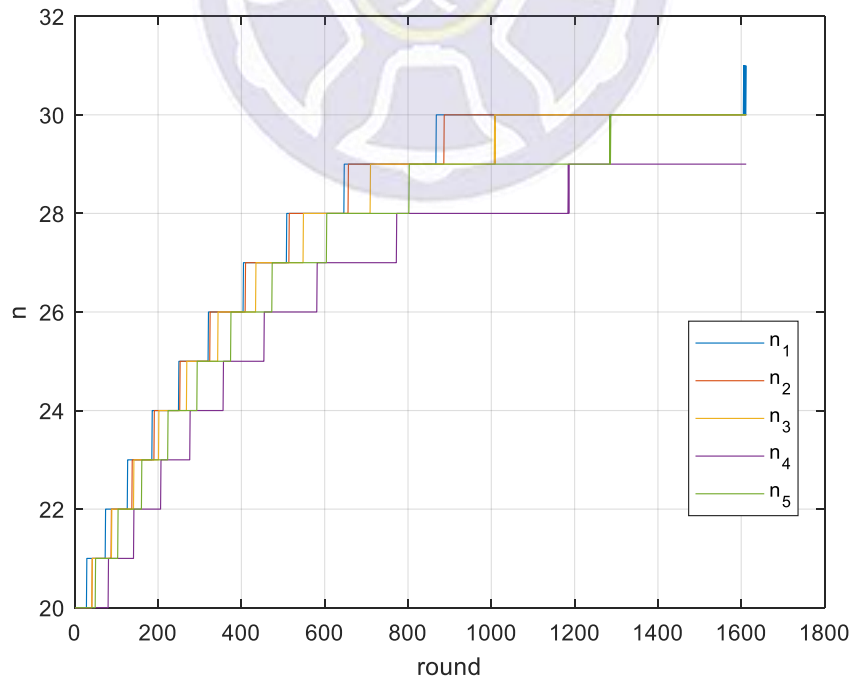


圖 4-4： n_k 的演進過程。

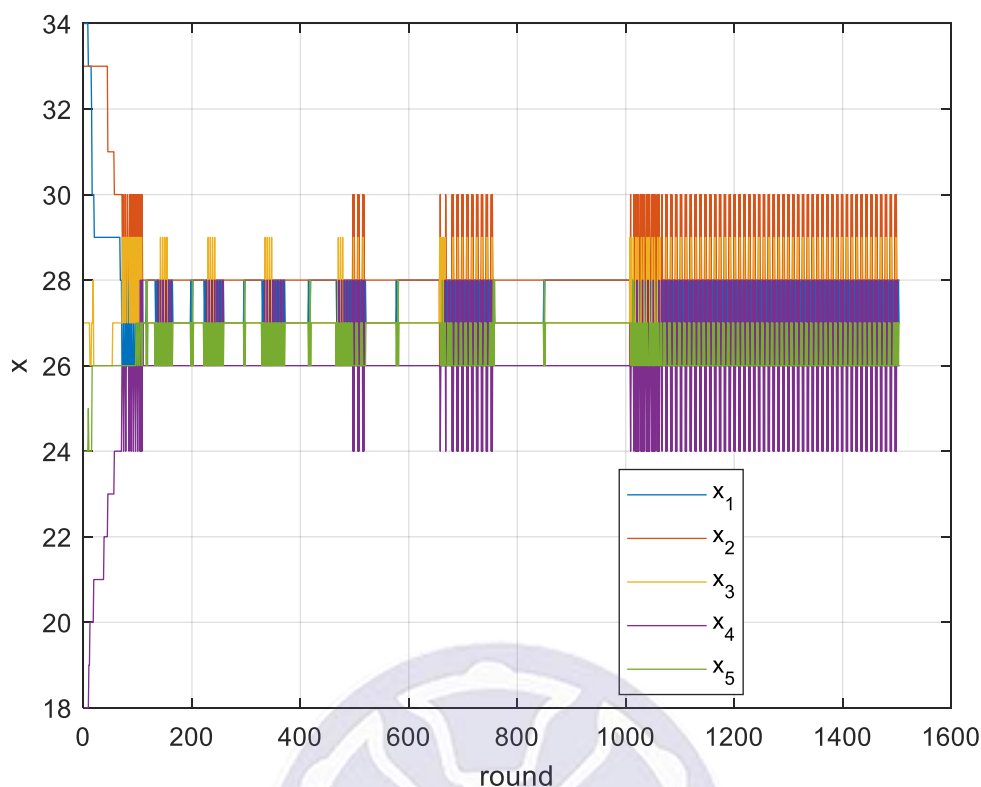


圖 4-5： $x_k = \sum_{i \in C} x_{ik}$ 的演進過程。

接著，圖4-3、圖4-4、圖4-5為各個參數的演進過程。根據(3.16)，也就是演算法1之終止條件，我們可對比 n_k 與 $\sum_{i \in C} x_{ik}$ 之大小，以發現 λ_k 是增加或是減少。由圖4-4與圖4-5，可以觀察到 n_k 一開始都小於 $\sum_{i \in C} x_{ik}$ ；因此，圖4-3中 λ_k 體現上升的趨勢。隨著 n_k 愈來愈大， n_k 與 $\sum_{i \in C} x_{ik}$ 之間差距愈來愈小，使得 λ_k 上升的趨勢也愈來愈緩；直到求得最佳解，也就是 $n_k = \sum_{i \in C} x_{ik}$ 後， λ_k 停止增加。

我們總結這部分的結果如下。在演算法1中，大部分用戶會與最近的AP進行連線。然而，為了避免用戶只與少數幾個AP進行連線，而導致資源分配失衡。少數用戶將不與速率最高的AP進行連線，而是選擇負載較小的AP進行連線。

4.3. 加入需求的限制

在3.3節中，我們探討加入用戶需求限制後，最佳化問題之處理如演算法2所示。在本節中，將使用MATLAB執行演算法2。除此之外，其結果將與演算法1的吞吐量進行比較，以展現我們所設計的演算法2相對於演算法1，可獲得多少增益。

4.3.1. 需求限制大小之影響

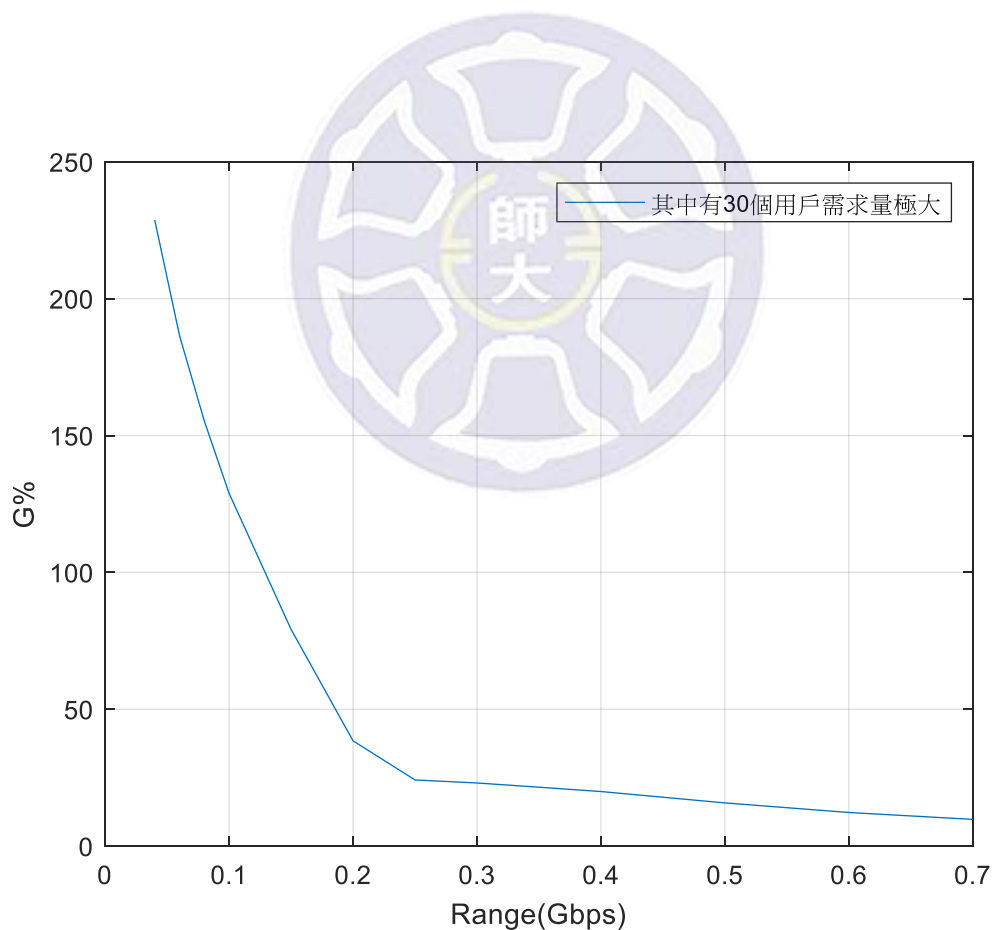


圖 4-6：當不同需求限制時，演算法2之增益。

在圖4.6中，系統設置了150個用戶，其中群組一包含120個用戶將受到需求限制影響；群組二則包含其餘30個用戶，因為需求極大，不受限制所影響。這樣的設定是因為我們想知道需求限制的範圍大小所造成的影響，也就是圖4.6中之Range。因此，將受到需求限制影響的用戶數量固定，以確認當執行演算法2時，Range所產生的資料量變化，不受其他因素影響。

在圖4.6中，縱軸表示演算法2所傳輸的資料量，相對於演算法1所傳輸的資料量來說，增加之百分比 $G(\%)$ 。橫軸表示需求限制的範圍大小Range，用戶的需求將在對應的數值範圍內進行均勻分布。從圖4.6中可看出在不同的Range數值，演算法2相對於演算法1，其總傳輸資料量的增加幅度。

如圖4.6所示，隨著Range的數值範圍愈小，群組一的120個用戶愈容易滿足。這使得演算法1所浪費的通道資源就愈多，我們所設計的演算法2之增益 G 就愈好，這與3.3節中的理論分析一致。換句話說，根據演算法1，即便該用戶的需求極小，AP依然分配固定比例的通道資源給該用戶；而在演算法2，當該用戶的需求極小，AP會在滿足該用戶需求後，把多出的通道資源分配給其他需求未滿足的用戶。因此，演算法2所獲得的總傳輸量，會比演算法1多。而且，隨著數值範圍Range愈小，演算法1所浪費的通道資源愈多，我們所設計的演算法2就愈優異。從圖4.6，我們觀察到，當 $\text{Range} = 0.1$ 均勻分布時，演算法2的總傳輸量可以是演算法1的兩倍以上，而根據其他的數值結果，最高甚至可以三倍以上。

4.3.2. 群組一數量多寡之影響

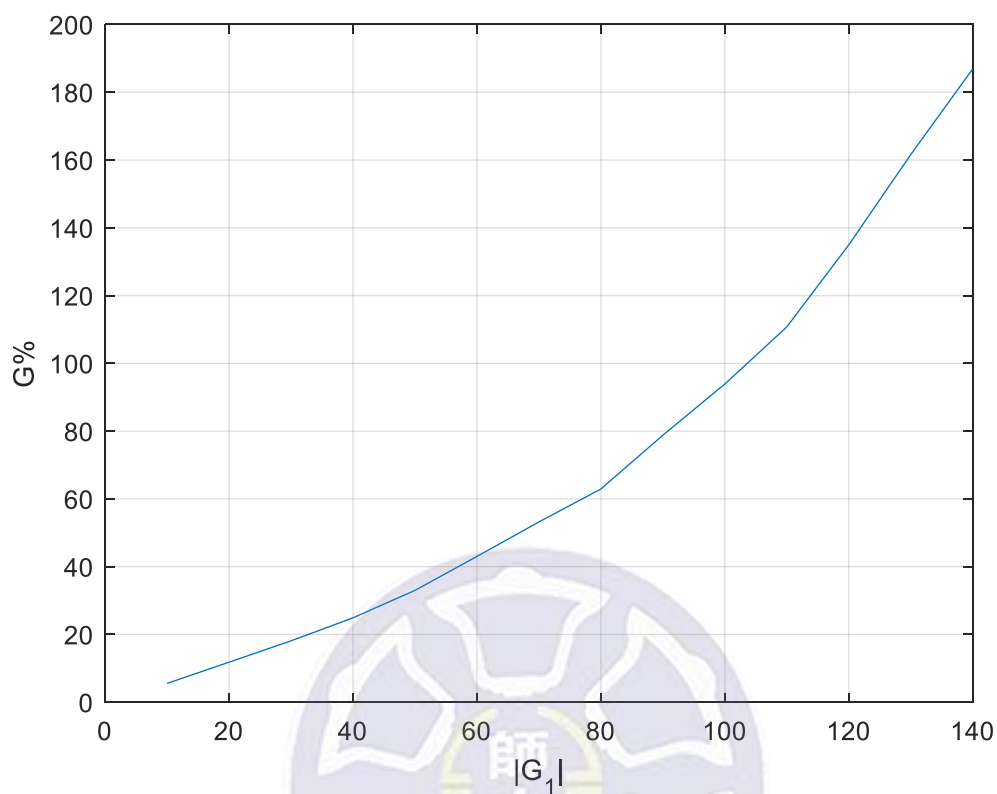
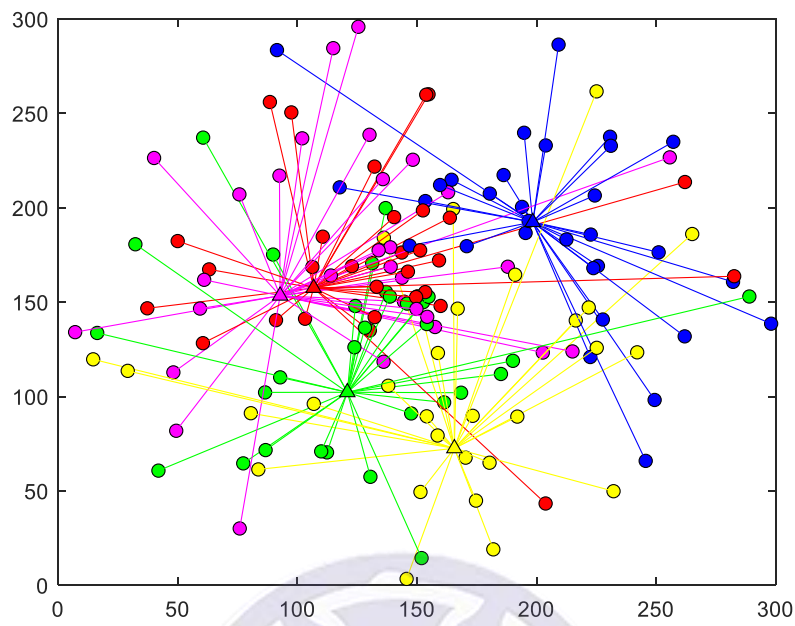


圖 4-7：不同的群組1用戶數量 $|G_1|$ ，演算法2相對於演算法1之增益 G ，其中 $\text{Range} = 0.1$ 。

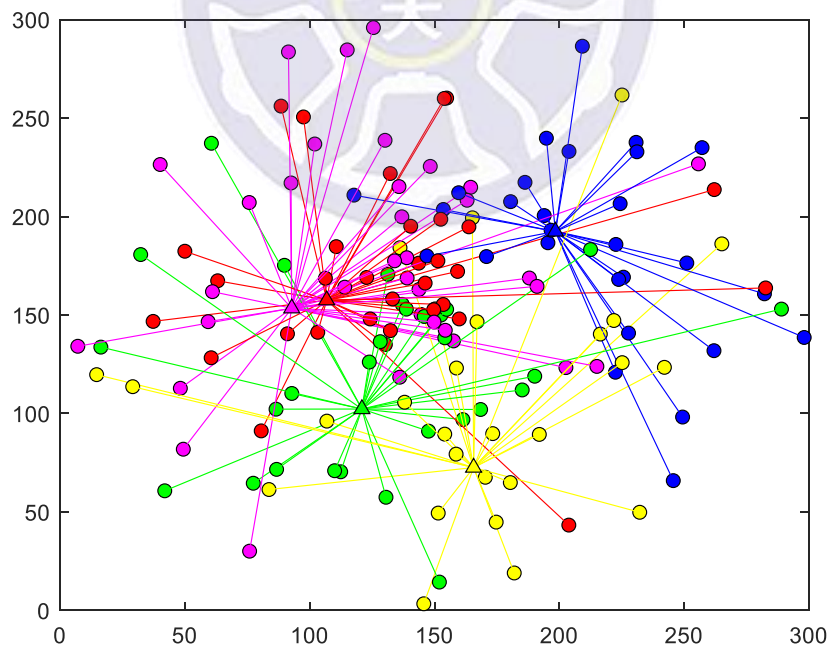
在圖4.7中，我們固定 $\text{Range} = 0.1$ 。縱軸為演算法2相對於演算法1總傳輸之增益 G ，而橫軸則為群組1之用戶數量，以 $|G_1|$ 表示。通過這樣設計，我們可以看出滿足需求的用戶數量之多寡，對增益所帶來的影響。

從圖4.7可以看出，隨著滿足需求的用戶數量愈來愈多，或 $|G_1|$ 愈大，演算法1所浪費的通道資源就愈多，而我們設計的演算法2就愈有利，這是符合預期的。相反地，我們觀察到，當可被滿足需求的用戶數量很少時，或 $|G_1|$ 很小時，演算法1所浪費的通道資源也很少，使演算法1與演算法2差別很少，這與3.3節中的理論分析也是一致。

4.4 模擬加入權重的演算法



(a)



(b)

圖 4.8：AP 與用戶的連線分布圖；(a)無權重時；(b)有權重時；

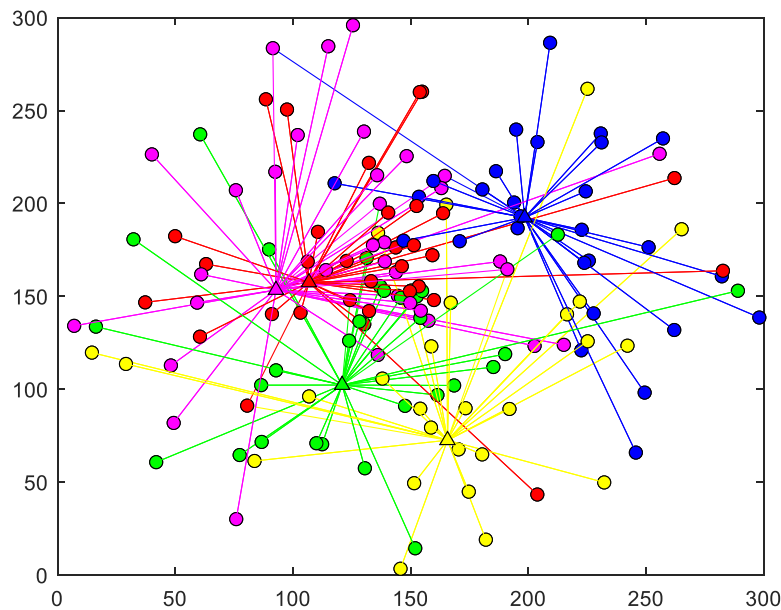


圖 4.9：將兩個結果重疊之連線分布圖。

考慮用戶需求時，除了需求限制外，還可以根據用戶的需求，給予不同的權重 w_i 。因此，在3.4節中，我們將最佳化問題改為(3.15)，並提出演算法3以求解。在本節中，我們分別對加入權重後之(3.15)與沒有權重之(3.2)進行模擬，或演算法1與演算法3，進行模擬，其結果分別如圖4-8與4-9所示。將兩個演算法之模擬結果進行對比，其討論如後。

4.4.1. 演算法1與演算法3之比較

首先，圖4-9為兩個演算法執行結果之重疊顯示。可以發現，不管有沒有權重，用戶連線的選擇幾乎相同。只有少數用戶選擇不同的AP進行連線。換句話說，即便(3.29)之 n'_k 為用戶權重之總和，而(3.13)之 n_k 為用戶數量，兩者用戶選擇的AP依然趨向相同。但是，兩者的通道資源分配 y_{ik} 卻不相同。根據(3.9)，在演算法1中，AP分配給每位連線用戶的通道資源皆相同；然而，根據(3.25)，在演算法3中，AP分配給每位用戶的通道資源，會依照權重大小

比例進行分配。如此，儘管在連線分布少有改變的情況下，可依照權重的方
式，呼應用戶的需求。

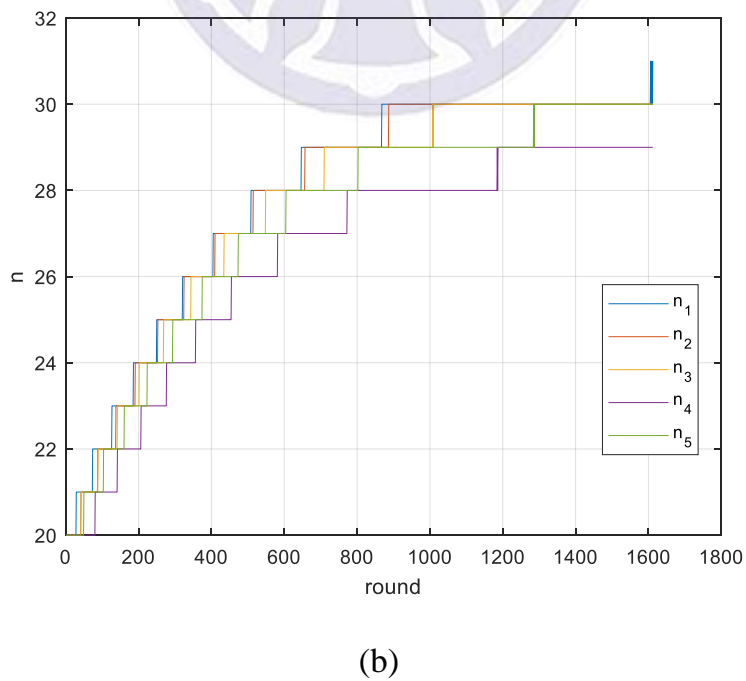
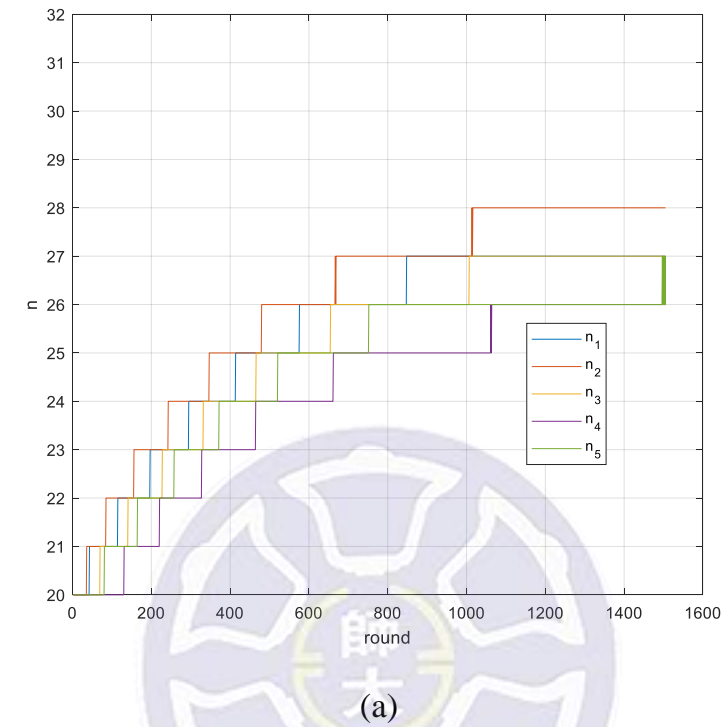
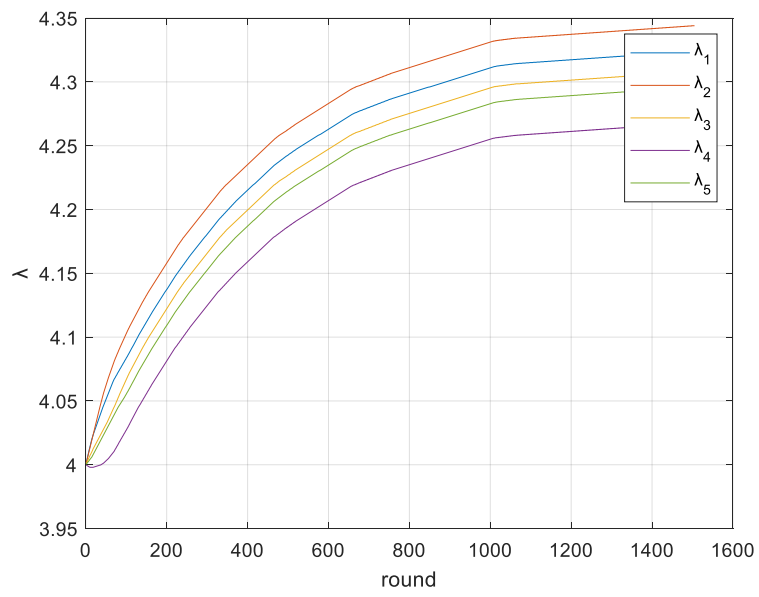


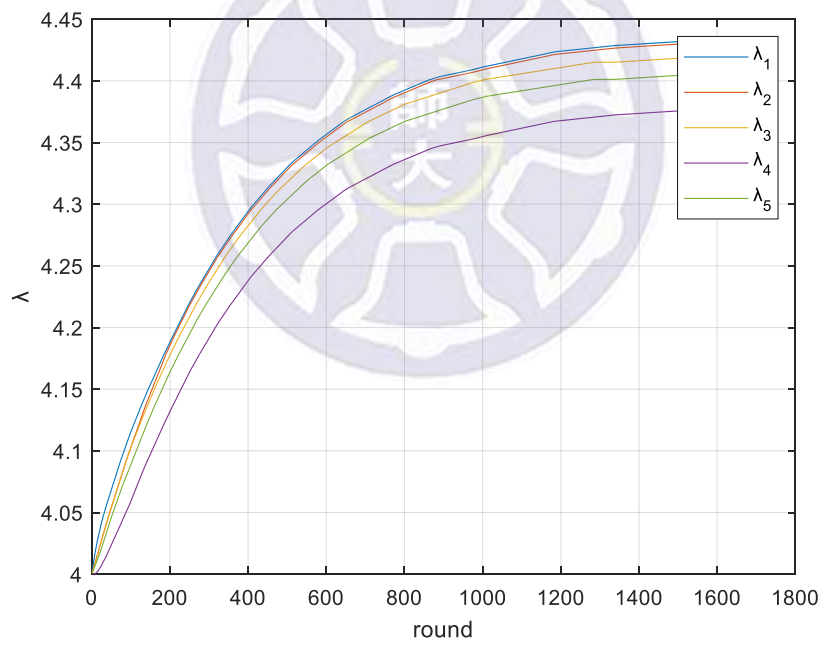
圖 4-10： n_k 的演進過程：(a)有權重時；(b)無權重時。

其次，根據演算法3，類似於4.2.1節，只有終止條件 $n'_k - \sum_{i \in C} w_i x_{ik} = 0$ 時，才有最佳解，其中，根據(3.29b)， λ_k 會影響 n'_k 的變化，即 $n'_k = e^{(\lambda_k - 1)}$ ，也就是說， λ_k 愈大， n'_k 愈大。根據(3.29c)，其各維度的斜率為 $w_i(b_{ik} - \lambda_k)$ ，也就是說， λ_k 愈大， $w_i(b_{ik} - \lambda_k)$ 愈小，與AP k連線的用戶愈少，也就是 $\sum_{i \in C} w_i x_{ik}$ 愈小，這同樣是一個權衡的過程。因此，根據模擬結果，類似於4.2.2節，由於 b_{ik} 之數值依然在22上下，可以發現與每個AP連線的用戶權重之總和會趨向相同。

最後，我們比較兩個演算法的演進過程。首先， λ_k 之初始值皆為4。觀察圖4-10與4-11，可以發現 λ_k 之變化十分相似。因此，演算法1之特性，加上權重後的演算法3也同樣具備。換言之，隨著 n'_k 愈來愈大， n'_k 與權重之總和 $\sum_{i \in C} w_i x_{ik}$ 之間差距愈來愈小， λ_k 上升的趨勢也愈來愈緩。須注意演算法3的用戶權重總和小於用戶數量，使得演算法3之 λ_k 在最佳解之值，會小於演算法1之 λ_k 。



(a)



(b)

圖 4-11： λ_k 的演進過程：(a)有權重時；(b)無權重時。

第五章 結論

5G毫米波網路為有前景的無線通訊技術，能夠提供更高的傳輸速率與更低的延遲。因此，在本論文中，將探討5G毫米波網路中，用戶連線的最佳化問題，以解決AP的分配、用戶間的公平性、與系統吞吐量。在研究中，發現它可以改寫為多維分配問題，並利用對偶分解法進行求解。為此，須了解下列過程，以作為設計用戶連線之依據。首先，當用戶與AP間的距離是已知的，可透過統計模型，計算訊雜比 SNR，以取得所需的參數。接著，根據比例公平性原則，設計連線最佳化問題。最後，利用對偶分解法，設計對應的演算法。

此外，考慮每個用戶之需求，另提出兩種用戶連線的最佳化問題：(a) 加上用戶的需求限制，使AP能更針對性分配通道資源；(b) 對應用戶需求加上權重，使AP根據權重大小進行通道資源分配。

根據模擬結果，了解各種最佳化問題的演進過程，並發現當考慮用戶需求後，最佳化問題的顯著改善與吞吐量之顯著提升。

附錄A 算幾不等式(arithmetic-geometric mean inequality)[17]

證明算幾不等式如下：

$$\frac{a_1+a_2+\cdots+a_n}{n} \geq \sqrt[n]{a_1 a_2 \cdots a_n}$$

首先，我們證明當 $n = 2$ 時， $\frac{a_1+a_2}{2} \geq \sqrt{a_1 a_2}$ ，使用差值比較法，即：

$$\frac{a_1+a_2}{2} - \sqrt{a_1 a_2} = \frac{(\sqrt{a_1})^2 + (\sqrt{a_2})^2 - 2\sqrt{a_1 a_2}}{2} = \frac{(\sqrt{a_1} - \sqrt{a_2})^2}{2} \geq 0.$$

故

$$\frac{a_1+a_2}{2} \geq \sqrt{a_1 a_2}, \text{得證。}$$

其中等號成立於 $a_1 = a_2$ 。

接著，我們證明當 $n = 3$ 時， $\frac{a_1+a_2+a_3}{3} \geq \sqrt[3]{a_1 a_2 a_3}$ ，先令 $a_3 = \frac{a_3 + \sqrt[3]{a_1 a_2 a_3}}{2}$ ，

利用當 $n = 2$ 時的算幾不等式，

$$\frac{\frac{a_1+a_2}{2} + \frac{a_3 + \sqrt[3]{a_1 a_2 a_3}}{2}}{2} \geq \sqrt{\left(\frac{a_1+a_2}{2}\right) \left(\frac{a_3 + \sqrt[3]{a_1 a_2 a_3}}{2}\right)} \geq \sqrt{\sqrt{a_1 a_2} \times \sqrt{a_3 \sqrt[3]{a_1 a_2 a_3}}} = \sqrt[3]{a_1 a_2 a_3} \quad ,$$

左右兩式整理得 $\frac{a_1+a_2+a_3}{2} + \frac{\sqrt[3]{a_1 a_2 a_3}}{2} \geq 2\sqrt[3]{a_1 a_2 a_3}$ ，移項整理得

$$\frac{a_1+a_2+a_3}{3} \geq \sqrt[3]{a_1 a_2 a_3}, \text{得證。}$$

我們把上述方法稱為代換法，因此，我們在嘗試一次，證明當 $n = 4$ 時，

$\frac{a_1+a_2+a_3+a_4}{4} \geq \sqrt[4]{a_1 a_2 a_3 a_4}$ ，先令 $a_4 = \frac{a_4 + \sqrt[4]{a_1 a_2 a_3 a_4} + \sqrt[4]{a_1 a_2 a_3 a_4}}{3}$ ，利用當 $n = 2$ 與

$n = 3$ 時的算幾不等式，

$$\begin{aligned} \frac{\frac{a_1+a_2+a_3}{3} + \frac{a_4 + \sqrt[4]{a_1 a_2 a_3 a_4} + \sqrt[4]{a_1 a_2 a_3 a_4}}{3}}{2} &\geq \sqrt{\left(\frac{a_1+a_2+a_3}{3}\right) \left(\frac{a_4 + \sqrt[4]{a_1 a_2 a_3 a_4} + \sqrt[4]{a_1 a_2 a_3 a_4}}{3}\right)} \\ &\geq \sqrt{\sqrt[3]{a_1 a_2 a_3} \times \sqrt[3]{a_4 (\sqrt[4]{a_1 a_2 a_3 a_4})^2}} = \sqrt[4]{a_1 a_2 a_3 a_4}, \end{aligned}$$

左右兩式整理得 $\frac{a_1+a_2+a_3+a_4}{3} + \frac{2^4 \sqrt[4]{a_1 a_2 a_3 a_4}}{3} \geq 2^4 \sqrt[4]{a_1 a_2 a_3 a_4}$ ，移項整理得

$$\frac{a_1+a_2+a_3+a_4}{4} \geq \sqrt[4]{a_1 a_2 a_3 a_4}, \text{ 得證。}$$

經由上述兩次嘗試，最後我們根據數學歸納法將 $n = k$ 擴及到 $n = k + 1$ ，即利用當 $n = k$ 時的算幾不等式，

$$\begin{aligned} &\frac{\frac{a_1+a_2+\dots+a_k}{k} + \frac{a_{k+1} + \sqrt[2]{\frac{a_1 a_2 \dots a_{k+1}}{k}} + \dots + \sqrt[2]{\frac{a_1 a_2 \dots a_{k+1}}{k}}}{2}}{\frac{a_1+a_2+\dots+a_k}{k} \left(\frac{a_{k+1} + \sqrt[2]{\frac{a_1 a_2 \dots a_{k+1}}{k}} + \dots + \sqrt[2]{\frac{a_1 a_2 \dots a_{k+1}}{k}}}{k}\right)} \\ &\geq \sqrt{\left(\frac{a_1+a_2+\dots+a_k}{k}\right) \left(\frac{a_{k+1} + \sqrt[2]{\frac{a_1 a_2 \dots a_{k+1}}{k}} + \dots + \sqrt[2]{\frac{a_1 a_2 \dots a_{k+1}}{k}}}{k}\right)} \\ &\geq \sqrt{\left(\sqrt[k]{a_1 a_2 \dots a_k}\right) \left(\sqrt[k]{a_{k+1} \left(\sqrt[2]{\frac{a_1 a_2 \dots a_{k+1}}{k}}\right)^{k-1}}\right)} = \\ &\sqrt{a_1^{\frac{1}{k}\left(1+\frac{k-1}{k}\right)} a_2^{\frac{1}{k}\left(1+\frac{k-1}{k}\right)} \dots a_k^{\frac{1}{k}\left(1+\frac{k-1}{k}\right)} a_{k+1}^{\frac{1}{k}\left(1+\frac{k-1}{k}\right)}} = \sqrt[2]{a_1 a_2 \dots a_k a_{k+1}}. \end{aligned}$$

左右兩式整理得 $\frac{a_1+a_2+\dots+a_{k+1}}{k} + \frac{(k-1)^{k+1} \sqrt[2]{a_1 a_2 \dots a_{k+1}}}{k} \geq 2^{k+1} \sqrt[2]{a_1 a_2 \dots a_{k+1}}$ ，移項整理得

$$\frac{a_1+a_2+\dots+a_{k+1}}{k} \geq \sqrt[2]{a_1 a_2 \dots a_{k+1}}, \text{ 得證。}$$

所以我們可以證明算幾不等式如上。

附錄B 斯萊特條件(Slater condition)與強對偶性 (strong duality)[18]

在數學中，斯萊特的條件(Slater condition)是強對偶性(strong duality)的充分條件，其作法如下所述。

其考慮的最佳化問題如下：

$$\begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & Ax = b. \end{aligned}$$

其中 f_0, \dots, f_m 為凸函數。其中 $f_i(x)$ 為不等式限制， Ax 為等式限制。斯萊特的條件如下，如果上述的原始(primal)最佳化問題有嚴格的可行解，即有一個可行解 x^* 滿足 $f_i(x^*) < 0$ ， $Ax^* = b$ ，存在 $x^* \in \text{relint}(\mathcal{D})$ ($\text{relint}(\mathcal{D})$ 指的是定義域 \mathcal{D} 的相對內部)，則該原始(primal)最佳化問題具備強對偶性(strong duality)，該條件稱為斯萊特的條件(Slater condition)。

我們讓(3.11) $\times (-1)$ ，即可從求最大化改寫為求最小化，其中(3.11)與(3.5)之限制皆為線性等式，沒有不等式存在，根據斯萊特的條件，當我們滿足線性等式時，有嚴格的可行解使(3.11)與(3.13a)之強對偶性成立，即有相同的目標值。

其中強對偶性(strong duality)的原理如下，如果原始(primal)問題(對偶(dual)問題)有一個確定的最佳解，那對偶問題(原始問題)也有一個確定的最佳解，而且兩個最佳解對應的目標值相同。

附錄C 權重轉換

須注意當求解 $\ln(a_1 a_2 \dots a_n)$ 時，與求解 $2 \times \ln(a_1 a_2 \dots a_n)$ 所得之最佳解皆相同，即目標函數乘上一個常數不影響最佳解的改變。因此，即便權重 w_i 不為整數，我們依然可以把目標函數乘上一個常數，使權重變成整數後求最佳解，舉例如下：

假設只有1個AP與2個用戶進行連線，其中分別求 y_1, y_2 ，假設 $w_1 = 1.6$ ， $w_2 = 2.5$ ，根據(3.20)，求解目標函數為 $\max_{y>0} \log(y_1^{1.6} y_2^{2.5})$ 。其中權重不為整數。須注意 $\max_{y>0} \log(y_1^{1.6} y_2^{2.5})$ 與 $\max_{y>0} 10 \times \log(y_1^{1.6} y_2^{2.5})$ 之最佳解相同。因此，轉為求解 $\max_{y>0} 10 \times \log(y_1^{1.6} y_2^{2.5})$ ，進行整理，變成 $\max_{y>0} \log(y_1^{16} y_2^{25})$ ，其中權重 w_i 已變成整數，並可以以這個目標函數求解。根據(3.22)，可以求得 $y_1 = \frac{16}{16+25} = \frac{16}{41}$ ， $y_2 = \frac{25}{16+25} = \frac{25}{41}$ 。

相關文獻

- [1] 下世代 5G 通訊規格除了更快，還會帶來什麼改變？
<http://technews.tw/2017/02/17/what-is-5g-the-laymans-guide/>
- [2] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta, and R. W. Heath, “Modeling and analyzing millimeter wave cellular systems,” *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 403–430, Jan 2017.
- [3] H. Elshaer, M. N. Kulkarni, F. Boccardi, J. G. Andrews, and M. Dohler, “Downlink and uplink cell association with traditional macrocells and millimeter wave small cells,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6244–6258, Sept 2016.
- [4] T. S. Rappaport, G. R. M. Jr., M. K. Samimi, and S. Sun, “Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design,” *IEEE Trans. Communications*, vol. 63, no. 9, Sept. 2015.
- [5] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter wave channel modeling and cellular capacity evaluation,” *IEEE Jour. Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, 2014.
- [6] Y. Xu, H. Shokri-Ghadikolaei, and C. Fischione, “Distributed association and relaying with fairness in millimeter wave networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 7955–7970, Dec 2016.
- [7] S. Cacciapuoti, , “Mobility-Aware User Association for 5G mmWave Networks,” *IEEE Access*, pp. 21497 - 21507 , Sep. 2017
- [8] 5G 到底有哪些 4G 沒有的優點？這裡告訴你答案

<https://kknews.cc/zh-tw/car/b6yxy46.html>

- [9] S. Yoshioka†, S. Suyama, T. Okuyama, J. Mashino, and Y. Okumura, “Digital Beamforming Algorithm for 5G Low-SHFB and Massive MIMO with Intersite Coordination,” The 20th International Symposium on Wireless Personal Multimedia Communications (WPMC2017), pp. 470–475
- [10] 5G 來了！你不可不知的技術趨勢與標準
<https://udn.com/news/story/6903/3046161>
- [11] 5G 來勢洶洶，你知道和 4G 差別在哪嗎？
<http://www.stufftaiwan.com/2017/03/28/>
- [12] 5G - 維基百科，自由的百科全書 <https://zh.wikipedia.org/wiki/5G>
- [13] 5G 除了速度比 4G 快十倍 還有哪些提升？
<http://tech.sina.com.cn/mobile/n/n/2018-01-26/doc-ifyqwiqk8355220.shtml>
- [14] 【5G 科普】只要 9 張圖，看懂什麼是 5G
<https://dahetalk.com/2019/03/01/>
- [15] L. Daniel, K. Narayanan, “Congestion Control 2: Utility, Fairness and Optimization in Resource Allocation,” Mathematical Modelling for Computer Networks- Part I , pp.2-1 – 2-10 , March 2013
- [16] Y. Bejerano, S.-J. Han, L. Li, “Fairness and Load Balancing in Wireless LANs Using Association Control,” *IEEE/Acmtransactions on Networking*, VOL. 15, NO. 3, JUNE 2007
- [17] 算 幾 不 等 式 的 證 明 (II) 與 (III)
<http://math.ntnu.edu.tw/~horng/letter/hpm1701.pdf>
- [18] 卡 羅 需 — 庫 恩 — 塔 克 條 件
<https://zh.wikipedia.org/wiki/>

自 傳

我出生在台北市，居住在新北市，身高 178 公分，體重 78 公斤，我的家人有媽媽、爸爸、弟弟。從小到大，我的家人都支持我。讓我學習多種才藝，包括畫畫、吹樂器等。

在大學階段，我原本上的是成功大學物理系，但由於後來對物理系不感興趣且因為在大學太過頹廢，導致被學校退學，後來改過自新，選擇電機系作為我的志向，轉學到輔仁大學。專題是學習怎麼設計電路使利用真空管的電路變好聽，其中運用到大學學的電子學、電路學等課程。

在研究所階段，可以發現與大學時差別甚大，起初我非常不適應，但隨著指導教授的指導，我漸漸的體會到如何獨自解決問題，學到很多東西，開始有獨立思考問題的能力，也具備了研究生的能力。

學 術 研 究

- [1] J.-J. Huang, and Y.-H. Ciou, “The Analysis of Rehealing Delay for a Sparse VANET with Unequal Traffic,” *ICUFN 2019.*, July. 2019.