

南 華 大 學

資 訊 管 理 研 究 所

碩 士 論 文

在行動環境中之可重複的資料廣播問題

Broadcasting with Data Replication in Mobile  
Environments

研 究 生： 王子修

指 導 教 授： 吳光閔 博士

中 華 民 國 九 十 四 年 六 月

南 華 大 學  
資 訊 管 理 學 系(所)  
碩 士 學 位 論 文

在行動環境中之可重複的資料廣播問題  
Broadcasting with Data Replication in Mobile Environments

研究生：王子修

經考試合格特此證明

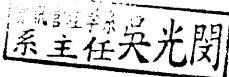
口試委員：\_\_\_\_\_

王碧斌

吳光閔

楊慶裕

指導教授：吳光閔

所 長：

口試日期：中華民國 九十四年五月 二十七 日

## 誌 謝

盛夏初至，又逢鳳凰花開時節，心中充滿離情，進入南華已快屆滿兩年，回想剛進學校之時，對於學習新知識總是充滿熱情與衝勁，但是學習這條路是非常艱辛的，必須放下很多的事務，心無旁騖全心全力的投入，更必須付出時間與精神，如果是職場工作者更必須犧牲部份工作時間，為人父母者更必須犧牲很多家庭生活及與親子相處的時間，這些都是生命中最重要的事，但當我正享受畢業滋味以及接收豐碩成果的時候，我並沒有忘記有一群人，你們正默默為我付出，提供我必要的支援與協助，陪伴我走過這段學習的日子，讓我兩年的學生生活毫無牽絆，並且能夠快樂的學習，在此謹誌了表我衷心的謝意。感謝主，感謝媽媽佳境，為我禱告讓我出入平安，內人怡君勤奮持家，一肩扛起打理家務重任，並且悉心照顧媽媽、女兒以琳以及兒子節宇，吳所長光閔對於研究方法的啟發及論文寫作的精髓更是傾囊相授，口試委員王昌斌主任、楊慶裕老師更對論文不吝指正，讓學生獲益良多，鐘國貴老師、邱宏彬老師、林豐智老師及許聖傑老師幫助我課業上的學習與解答困惑，林峻鋒老師指導程式語言撰寫技巧，李秀芬老師教導有關英文的閱讀學習方法，研究夥伴文成、新春不時給予指導鼓勵，另外如智郎、婉玉、明峰、書萍、于紋、峰進、國棟、峻維、沛錡、賜榮、延昇、祐全、慶春、永諒、明德、銘宗等，謝謝你們，沒有你們一路相伴以及真誠的付出，我想學習會更加辛苦。今日本論文得以順利完成，我將這份喜悅與榮耀獻給各位，期望能在不久的將來再次相聚，並獻上我衷心的感謝與祝福，謝謝你們。

子修敬書

2005.06 於南華大學

# 在行動環境中之可重複的資料廣播問題

學生：王子修

指導教授：吳光閔

南 華 大 學 資 訊 管 理 學 系 碩 士 班

## 摘 要

行動環境中礙於頻寬不足，用戶端可藉由無線廣播的方式取得所需資訊，而一組好的廣播序列可以幫助使用者降低資料存取時間。本論文藉由分析所有用戶端所需要的資料物件，依熱門程度插入至廣播週期中，再經由模擬退火法的協助，找尋一組較佳的廣播序列，用來快速滿足用戶端需求。經實驗證明本研究方法，在廣播週期中將熱門資料物件重複廣播，的確可以有效降低用戶端的平均接收時間，降低率約在 70%~85%，並且發現重複過多資料物件會導致廣播週期增長，而造成反效果。

關鍵字：行動環境，無線通訊，無線廣播，模擬退火法。

# Broadcasting with Data Replication in Mobile Environments

Student : Tzu-Hsiu Wang

Advisor : Dr. Guang-Ming Wu

Department of Information Management  
The M.B.A. Program  
Nan-Hua University

## ABSTRACT

Due to the restriction of bandwidth in mobile environments, users have to access their wanted data by mobile device with the help of broadcasting technology. Intuitively, a good scheduling algorithm is capable of reducing the access time. In this thesis, we analyzed the most popular data objects, we insert that to the broadcast cycle by their fever first, then we search for a better broadcasting schedule to satisfy users quickly by way of simulated annealing. Experimental results show that our method is effective in reducing the average receiving time of users by repeated broadcasting of the hot data objects in a broadcast cycle. The decreasing rates are between 70% and 85%. Besides, reiterated data objects would get longer broadcast cycle contrary.

Keywords : mobile environment, wireless communication, data broadcast, simulated annealing.

# 目 錄

書名頁 .....	i
博碩士論文授權書.....	ii
論文指導教授推荐書.....	iii
論文口試合格證明.....	iv
誌謝 .....	v
中文摘要 .....	vi
英文摘要 .....	vii
目錄 .....	viii
圖目錄 .....	ix
表目錄 .....	x
第一章 序論 .....	1
第一節 研究背景與動機 .....	1
第二節 研究目標與限制 .....	8
第三節 論文架構 .....	13
第二章 問題描述 .....	14
第一節 廣播環境與瓶頸 .....	14
第二節 單調式效能量測 .....	16
第三節 非單調式效能量測 .....	19
第三章 我們的演算法 .....	28
第一節 熱門資料物件的處理 .....	28
第二節 廣播資料物件的處理 .....	30
第三節 模擬退火法 .....	32
第四章 實驗結果 .....	38
第一節 實驗環境 .....	38
第二節 實驗資料及過程 .....	39
第三節 結果分析 .....	43
第五章 結論與未來研究方向 .....	51
參考文獻 .....	53
附錄一 .....	58
附錄二 .....	61
附錄三 .....	67

# 圖 目 錄

圖一	無線通訊環境基本架構圖	2
圖二	模擬退火法組態圖	12
圖三	無線廣播架構圖	15
圖四	無線廣播中用戶端接收時間	18
圖五	查詢資料集配置	19
圖六	廣播週期中重複資料物件與距離量測關係圖	21
圖七	週期中重複資料物件距離量測圖	22
圖八	週期中重複資料物件距離量測圖	23
圖九	週期中重複資料物件距離量測圖	24
圖十	週期中重複資料物件距離量測圖	25
圖十一	週期中重複資料物件距離量測圖	26
圖十二	模擬退火法描述概圖	32
圖十三	擾動方法 (op1)	34
圖十四	擾動方法 (op2)	35
圖十五	實驗過程及步驟	42
圖十六	指數資料分佈下重複與不重複資料物件關係圖	44
圖十七	常態資料分佈下重複與不重複資料物件關係圖	46

# 表 目 錄

表一	符號定義	14
表二	廣播資料物件熱門程度統計表	29
表三	排序後熱門資料物件表	30
表四	實驗程式與檔案輸出入情形	40
表五	指數型態資料分佈實驗數據	48
表六	常態型態資料分佈實驗數據	49



# 第一章 序論

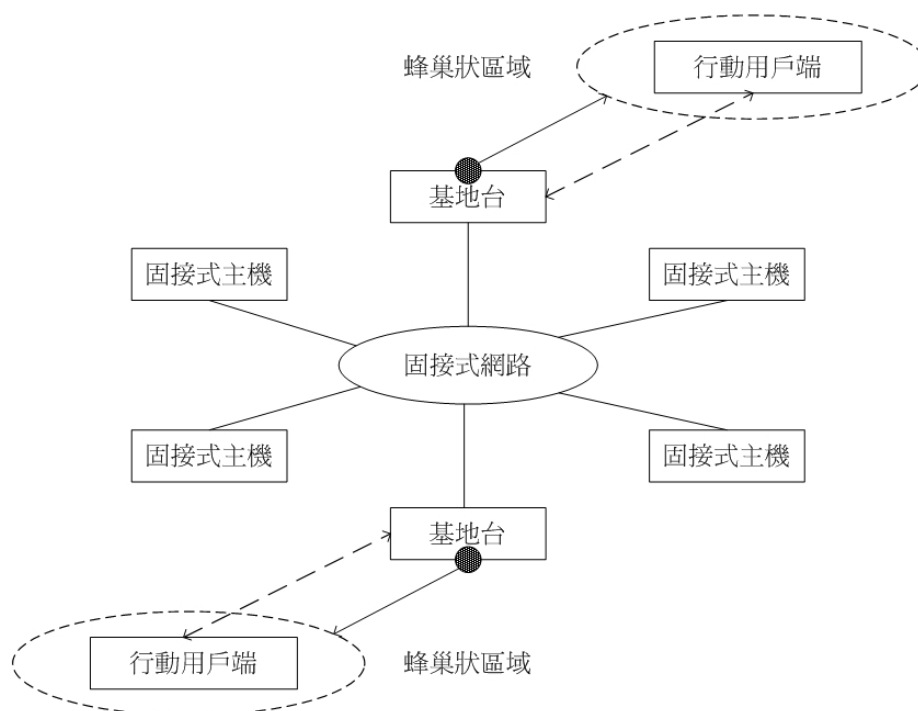
## 第一節 研究背景與動機

人與人之間藉由彼此的相互溝通，達到滿足彼此的需求與好奇，隨著人類歷史的演進，溝通的方式與媒介，不斷的推陳出新，拜電腦科技的高度發展所賜及網際網路的盛行，人類相互間的溝通以及資訊的傳遞，在網路中突破了空間與時間的藩籬，人類的生活與習慣因網路的普及化及相關商業應用的發展，產生了重大的變革。

電子商務( Electronic Commerce; EC )的高度發展對經濟活動和社會環境影響深遠，主要因素在於電子商務具有高效率、快速交易及無國界的特質，這些都是有助於全球化商業電子活動的推展。電子商務對經濟的影響，主要是由於網際網路的出現，改變了傳統產業貨物流通及資訊傳遞的方式，進而改變廠商組織型態和市場結構，降低交易成本，提升資源的使用效率。利用網路搜尋系統，取代傳統勞動密集的市場行銷和分配通路，降低整體經濟的交易成本，進而提高了經濟成長力[4]。對於網際網路而言，透過網路進行交易不但能節省成本，也不必受時間與空間的限制，可提供全年無休的服務，不但促使了傳統商業的經營模式快速轉變外，同時也帶來了無限商機。

當全球資訊基礎建設完成後，地球的資訊資源將被緊密地結合在一起，沒有地理及時間方面的障礙，所有人都能透過簡單的設備連接上這個網路，並享用這些公共的世界資源，所有的組織與企業均想利用這些

資源，幫助企業自身的成長；故網際網路的熱潮，已為企業界帶來相當大的商機，對於未來商業市場及科技環境變動的速度會愈更加迅速，以往以人力遞給文件的方式，勢必無法跟上時代變動的腳步。企業採用電子商務帶來下面三個主要利益，改善顧客服務、改善與供應商之間的關係以及增加股東與投資人的紅利，故全球商家都想在這新市場佔據一席之地[1]。



圖一 無線通訊環境基本架構

為了擺脫網路系統的束縛，實現與真實生活相互結合，行動環境的發展除了提供使用者可以隨時隨地使用網路提供的各項服務功能外，對使用者而言，無非能在適當時間或者適當地點，毫無拘束地擷取所需資訊，滿足個人對於資訊的迫切需求。伴隨科技的日新月異，無線通訊(wireless communication)的使用已相當普遍，在無線網路(Wireless LAN)中，因為無線網路的移動性和方便性，使得無線網路的應用例如PDA(Personal Digital Assist)、行動電話(Mobile Phone)及筆記型電腦(Notebook)的使用，截至目前已成為人類生活的一部份，且扮演極為重要的角色。其基本架構如圖一所示。主要成員包括有行動客戶端(mobile client)、基地台(base station)及固接式主機(fixed host)。基地台透過固接式網路(fixed network)與固接式主機連結，固接式主機提供基地台資料庫管理查詢等動作，而行動客戶端則透過無線通訊頻道與基地台溝通[2]。

人們運用行動電話收發簡訊，但是這些基礎功能已無法滿足一般消費者需求。如何滿足特定族群的需求，進行系統的即時播放，提供最新最完整的資訊是通訊廠商吸引客戶的主要方式。例如將隨時在波動的股市資訊，即時的傳送至投資人的小型接收器上，投資人只需將所要的各類股的資訊設定好，接收器會定時運用系統化的方式進行目前股價的更新，提供使用者極為方便的資訊接收方式。另外如先進的衛星導航系統也是運用廣播的方式進行資料的傳送，此種先進的系統已被廣泛的運用在交通資訊管理系統上，例如飛機的飛

行航線導引、汽車行車路線導引或是觀光旅遊導覽位置參考等。

就目前行動設備的應用而言，逐漸由個人職場所需資訊記錄(行事曆、備忘錄)、網路服務(郵件收發、資訊瀏覽)，轉而結合生活與休閒需求，例如全方位數位家庭多媒體軟體權威力新國際，繼日前推出業界第一套支援新一代Intel® 迅馳™(Centrino™)行動運算技術，代號Sonoma的即時無線錄影數位娛樂產品Presto! WVR (Wireless Video Recorder)後，重新整合公司資源並獨創無線廣播技術，完成多媒體無線影音廣播系統Presto! WMS (Wireless Multimedia System)，用戶端透過筆記型電腦或無線網卡，Presto! WVR可將影音多媒體資料，經由接收器，同時以一對一或者以一對多的廣播方式，將資料內容傳送到類比、數位電視或液晶(LCD)螢幕上，使用者便可在家中隨處輕鬆欣賞影片，甚至可直接在電腦上編輯錄製的影片內容或製作成影音光碟，故有關無線網路的應用與研究，成為當前網路發展最重要的一環。

上述行動環境的議題有關無線廣播(Wireless Broadcast)就是一個例子，透過廣播裝置及使用者接收設備，資訊提供者可以隨時隨地提供使用者所需資訊，使用者對於資訊的取得，不再受限於任何時間與空間，只需備妥行動接收設備，就可以滿足生活所需，建置這樣的一個環境，大大的改變了人們的活動習慣及快速滿足對於資訊的渴望。

無線區域網路架構依連接方式不同主要可區分為，對等式無線區域網路(Ad-Hoc Network)與基礎式無線網路(Infrastructure Network)等

兩種。Ad-Hoc是一種點對點的傳輸模式，在802.11無線網路架構中，可以讓單一設備或工作站直接進行點對點溝通，而且不需經由無線AP (Access point)。Infrastructure Network主要區分單播 (Unicast)、群播 (Multicast)及廣播 (Broadcast)等三種形式。單播在網路上經常被使用，它是一種在點對點的網路上進行一對一串流播送的方式，在單播環境中，由使用端主動到網路上取得資格或授權，而檔案則在被訂閱或點播之後，由伺服器以一對一的方式傳送給每一個訂閱的用戶端。而群播同樣在一般網路廣播媒體中被普遍採用，它採用單對多的傳送方式，在點對點網路上做一對多的串流播送，在群播環境中，資料與節目由發送端同步送到許多的接收點，並透過較大頻寬以資源共享方式來節省網路資源。廣播採用的是點對多的單向傳輸，使用者不需提出要求，它會在廣播網路上對外進行資訊及節目的傳送。因此，在這樣的廣播環境中，接收端是比較無知與被動的，就像一般傳統無線電視台的節目，不論用戶是否打開電視都在進行節目訊號的傳送，它就是屬於這種方式。在廣播的環境之中，我們稱存在於廣播頻道中的資料為資料物件 (Data Object) 或是稱之為資料桶 (data bucket) 經由廣播系統分析後，產生了一組廣播序列，並且被依序放置於廣播頻道 (Broadcast Channel) 上進行廣播傳送，接收用戶端經由持續監聽廣播頻道，由其中擷取相關所需資訊。一般而言，無線廣播網路的服務主要採取下列兩種形式呈現，一種是拉模式(Pull Mode)[5]，另一種則是推

模式(Push Mode)。

Pull mode亦稱為要求式廣播（ On-Demand Broadcast ），用戶端需要主動到網路上擷取到他所需要的內容，廣播系統於收到使用者提出需求之後，各項資訊種類透過演算法計算後，將演算法所認定優先廣播的資料物件群放置到廣播頻道上，並且進行廣播。因資料與節目只有在用戶端即時要求下，才會進行播放。如果純粹使用Pull mode進行廣播，假定出現大量使用者的情況時，廣播系統無法在極短時間內，對用戶端所需求的資訊種類運用演算法進行廣播判斷，因此會造成廣播週期中部分時間區塊（ time slot ）的閒置狀態，網路廣播頻寬（ bandwidth ）並因此未能有效完全運用，但如果在使用者大量出現時廣播系統轉換為Push mode，則可運用其優點，由歷史資料庫所產生的廣播排程進行廣播，如此就不會被使用者數量多寡所影響，進而改善了網路頻寬閒置的問題。

Push mode主要是由廣播系統統計歷史資料物件被需求的相關情形，依照各個資料物件被擷取的頻率（ Access frequency ），並且進行演算程序後，產生一組資料物件的廣播排程，在固定時間內，這組廣播排程資料物件有規律的放置於廣播頻道之中，形成一組廣播循環（ Broadcast Cycle ）序列，用戶端只需聆聽並進行所需資料的接收，它不需用戶透過網路提出要求，系統本身就會將資訊主動往外傳送的服務模式。也就是說，它是一種不論用戶端是否需要，都會發送訊號

與資料的方式。相同地若單獨使用Push mode進行廣播，某些用戶所需要的是較為冷門的資料(Cold data)，會因為這些冷門資料在歷史資料庫被需求的機率並不高，因此可能在眾多排程演算法中被輕忽，甚至無法放入排程之中，造成冷門資料的用戶端等待時間過長或者毫無下文。若可以經由Pull mode以及Push mode兩種模式混和使用，當頻道負載重或者使用者數量過多的時候使用Push mode，而在負載輕且使用者少的情況時切換使用Pull mode，如此便能更有效率的滿足用戶端需求，並且改善廣播週期可能的閒置狀態。這類的廣播技術常會使用一個判斷函式，決定在廣播頻道負載輕、重時，系統可以切換使用Pull mode或Push mode的演算法，使用此種混合方式的廣播技術，可以有效互補兩種廣播模式的缺點並且增加廣播效率[31]。Pull mode以及Push mode兩種模式的適用時機以及如何取得平衡點，是一個相當重要的議題[7,14,28,29]。

本研究論文主要針對Infrastructure Network中Push mode無線廣播系統環境，期望應用相關歷史資料物件的需求頻率，透過演算法的協助，幫助我們找到好的資料物件廣播序列，讓用戶端在有限頻寬中能夠更有效率的接收所需資料。

## 第二節 研究目標與限制

藉由無線廣播技術的發展，人類透過無線通訊不管身處任何時間、任何地方均能毫無限制的取得所需資訊，不同於傳統有線網路，無線網路環境被考慮主要有以下兩部分。

第一是無線網路頻寬和行動設備能源需要的限制，行動設備端其能源消耗一般分成兩種模式，活動模式(active mode)及睡眠模式(doze mode)，當行動用戶端處於活動模式時，其功率消耗為處於睡眠模式時的數千倍。行動用戶端於廣播頻道中監聽資料時，是依廣播協定(broadcast protocol)，決定何時該進入睡眠模式，又何時該進入活動模式，於廣播循環週期中讀取資訊，直到完整取得所要資訊為止[19,22]。行動環境通訊相關研究領域中，用來評估無線廣播機制主要有兩個關鍵參數，一為接收時間(access time)，即行動用戶端提出一個查詢要求開始，直到完全收到所需資料物件為止，所需耗費的時間總和稱之；二為喚起時間(waking time)，即用戶端從進入省電模式到被喚醒開始接收訊息的這段時間，兩參數主要影響行動用戶端能源消耗情形，此類相關的研究如使用快取及非一致性廣播來減少接收時間[5,6,27]，有些則採用索引法及雜湊法[10,16,20,21,23,24,25]將資料廣播中的額外資訊 ( additional information) 告知資訊接收端下次資料擷取時間為何時，藉此降低喚起時間，達到行動客戶端省電的目的。[26]在Pull mode廣播環境下，排除索引法可能導致廣播週期過長以及行動主機 ( Mobile



Hosts;MHs ) 反應時間。

第二是不對稱的環境，從能源消耗的觀點來說，移動式行動裝置發送資料比接收資料需較高之成本。由於存在這些問題的緣故，在無線環境中傳統的需求回應廣播系統不再只是滿足資訊的傳送，因此資訊在廣播頻道上傳送的方法，便成為此領域中為眾人所感興趣之研究議題[15]。如 [5,13,15,22,30]有效組織廣播資料序列，用來降低用戶端接收時間。[5,6] 對每一資料物件的使用喜好及資料物件傳送次數間的差異，較受歡迎的資料物件重複被廣播等，這方式會增加廣播週期的長度，對使用較少熱門資料物件的用戶端而言，無疑增加它的平均接收時間，上述方法並沒有考慮兩資料物件的關係。相對於上段所述行動裝置能源消耗的問題並非本論文探討的核心，我們主要還是在研究如何利用熱門資料物件並且改善無線廣播資料配置方式，來縮短行動用戶端平均接收資料時間。

就現實生活面，用戶端所需要的資料物件種類、大小並不一定相等，無線廣播研究中，資料長度較大的資料物件，相對其廣播所需花費的時間也一定較長。為使研究環境單純化，我們假設資料物件的長度均相同，也就是在廣播頻道中的每個單位時間廣播出來的資料物件或是資料頁大小皆相同。依使用者的需求數目來區分，當使用者提出了一個只有包含一個資料物件的需求給伺服器端時，我們稱之為單一要求 (single query) 的廣播模式；相反的使用者提出了一個包含二個以上 (含二個) 資料物件的需求給伺服器端時，我們稱之為多重

要求 ( complex query ) 的廣播模式。

我們假設每個資料物件大小相同，伺服器端參照以往資料物件被需求的查詢頻率，運用學者研究的廣播排程技術，將這些資料物件作最佳化排列，並且運用廣播循環的方式進行循序播放。廣播通道又區分單一通道及多通道，[9]即是多通道廣播環境下針對每一廣播通道進行資料最佳化配置。對於廣播排程資料物件擺放技術的研究，進一步分成廣播序列中資料物件不重複的單調式 ( flat ) [5]廣播方式，另外資料物件依照出現頻率多寡加以分析後，將多數用戶端喜愛的熱門資料物件重複插入廣播序列中，就是單一週期允許資料物件重複的非單調式 ( non-flat ) [5]廣播方式。

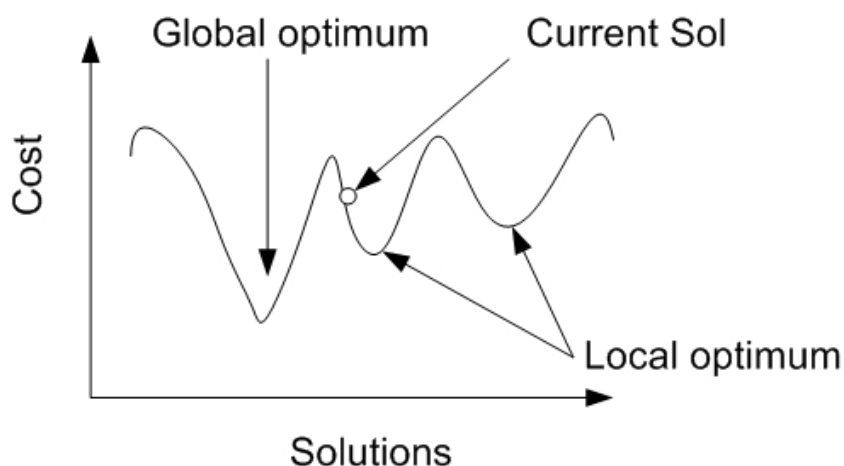
較早的研究，以單一要求及單調式的廣播模式為開始，實驗環境也最為單純，需要考慮的狀況也是最少，隨後多重要求及非單調式的廣播模式[8,17]的相關研究逐漸受到重視。就現實生活環境而言，每一位使用者提出多個資訊種類的要求是比較合乎常理的。因此開始討論在多重要求下的廣播模式，每個使用者均可能要求多個資訊種類，然而為廣大使用者所偏愛的資料物件亦可能出現於單一廣播週期中，形成同一週期中資料物件重複的情況。另多時分工(Time Division Multiplexing; TMD)傳播環境架構[12]將資料物件配置到閒置的廣播頻道上。因此本篇研究重點在於多重要求及非單調式的廣播模式下進行，為求研究環境單純化，我們假設資料物件的長度是一致

的，每一用戶端則有不同需求的資訊種類組合，按照資訊種類被需求的頻率高低來進行統計排序，也就是了解每個資料物件受歡迎的程度，並且將受歡迎的熱門資料物件擺放至廣播序列中，運用相關邏輯演算法則，使得用戶端平均接收時間最小。

在我們的廣播資料庫發行方法之中所運用的資料庫型態屬於關聯式資料庫，而我們將關聯式資料庫中的每一個資料欄位，所包含的資料總和都稱之為一個資料集 (Dataset)，也可稱它為資料物件，在我們所討論廣播頻道中的資料廣播順序，所指的就是資料表中資料欄的相關順序。使用者端就是依據廣播伺服器所組成的廣播資料順序進行資料的擷取動作。但是資料的擷取會因廣播的順序而有差異，因此，為了避免用戶端等待資料的閒置時間過長，不同的廣播資料排列順序會影響到使用者端的平均接收時間，所以較佳的資料物件排列順序是我們在無線廣播中尋求的目標。

本論文使用模擬退火法(Simulated Annealing; SA), 將金屬合金的退火過程與廣播排程最佳化問題的求解過程予以類比，所衍生出一種求近似最佳解的演算法。模擬退火法的最大特徵在於它除了能夠接受較優的解之外，仍然給定一機率用以接受較差的解。此一機制運用在退火過程中，最初溫度 $T$ 值較大時，系統接受較差解的機率較高，隨著溫度 $T$ 的逐漸下降，較差的解的被接受的機率也跟著降低，最後當 $T$ 下降趨近於零值時，較差的解就不再會被接受，這個特徵使得模擬退火法不同於一些

鄰近搜尋法(Neighbor Search)，其具有跳出局部最佳解(local optimum solutions)，而達到全域最佳解(Global optimum solutions)的能力[3]，如圖二。



圖二 模擬退火法組態圖

本論文提出使用模擬退火法，幫助我們找到較佳的廣播序列，使行動用戶端能在最短時間內取得所需資料，無形中對於行動裝置能源的節約產生莫大幫助。在本文第四章中將介紹整個實驗的相關過程。但對於如何節省用戶端行動設備的能源需求，並非本篇論文主要研究方向，本研究的目的還是著重在降低接收時間，因為用戶端的接收時間降低了，無形中幫助節省了行動設備的能源損耗。我們在多重要求、單調式以及擷取資料無順序性的廣播環境下進行研究。隨意擺放受歡迎的資料物於單一廣播週期中，並且經由模擬退火法不斷的擾

動、降溫，透過成本函數的計算，將廣播序列導引至最佳狀況，模擬實驗中顯示了重複熱門資料物件於同一廣播週期中，對於縮短用戶端接收時間是有幫助的，但是隨著越來越多熱門程度不一的資料物件插入，使的單一週期長度變的過長，相對於多數用戶端等待接收的時間變長，而引發反效果，所以有限制數量的資料物件重複是絕對必要的。用戶端需求資料呈現指數分佈時，經由模擬退火法，總接收時間改善幅度可達70%~85%；另外在常態分布的模擬環境中，總接收時間的改善幅度亦在70%~81%之間。其結果對於行動用戶端降低平均接收時間是有明顯幫助的。

### 第三節 論文架構

本論文第二章將進行有關廣播環境所遇到的問題描述，並且詳細說明了單調式與非單調式效能量測的方法，便於實驗中做比較。第三章是我們的演算法介紹，主要說明熱門資料物件的統計方法、廣播資料物件的初步處理、運用模擬退火法找尋最佳的廣播排程以及相關演算法。第四章是實驗結果，主要說明實驗環境、實驗用資料的產生及整個實驗過程，並就實驗所獲得的結果加以分析探討。第五章是結論，說明我們未來期望研究的議題與方向。

## 第二章 問題描述

### 第一節 廣播環境與瓶頸

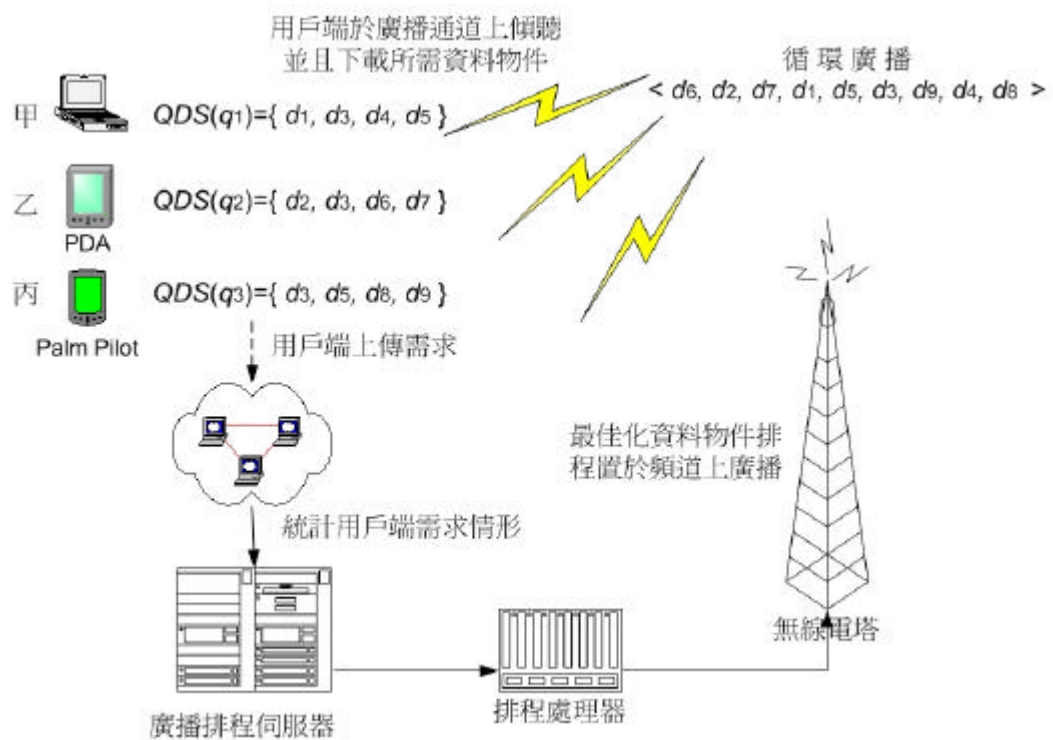
首先，我們在表一中提供相關符號的定義，這些符號將用於整篇論文中。

符號	意義
$d_i$ $ d_i $ $D$ $D_h$ $ D $ $B$	某一個被廣播的資料物件 一個資料物件之大小 所有被廣播資料物件的集合 經統計後熱門資料物件的集合 所有廣播所包含資料物件的數量 一個廣播週期的長度
$q_i$ $QDS(q_i)$ $ QDS(q_i) $ $freq(q_i)$ $f(d_i)$ $Q$ $ Q $	在廣播頻道中被提出的某個查詢 $q_i$ 所存取資料物件的集合 $q_i$ 所存取資料物件的數量 $q_i$ 之查詢頻率 $d_i$ 資料熱門程度 所有的查詢所成的集合 所有的查詢所成的數量
$s$	單調式廣播頻道中資料物件安排順序 非單調式廣播頻道中資料物件安排順序

表一 符號定義

大多數的用戶端，將用戶所需要的資料物件類別透過行動設備，傳送給廣播伺服器，廣播伺服器收到用戶端需求後，利用各種演算程序，找到最佳廣播資料物件的序列，然後將資料物件至於廣播通道上進行循環式廣播，用戶端透過行動裝置，從廣播頻道中截取所需要的資料物件，物件經由不斷的循環廣播，用來隨時更新用戶端資訊。

圖三說明了現今無線廣播環境的主要架構，用戶端甲、乙、丙個別所需資料物件分別為 $\{d_1, d_3, d_4, d_5\}$ 、 $\{d_2, d_3, d_6, d_7\}$ 以及 $\{d_3, d_5, d_8, d_9\}$ ，用戶端經由上傳需求，傳送所需要的資訊要求到廣播伺服器，伺服器進行邏輯演算後，產生較佳的廣播資料物件序列，



圖三 無線廣播架構圖

$s = \langle d_6, d_2, d_7, d_1, d_5, d_3, d_9, d_4, d_8 \rangle$ ，排程處理器依照廣播資料物件的順序，將資料物件傳送至廣播頻道上進行廣播，資料物件被重複在頻道上廣播循環排程。用戶端則藉由監聽廣播頻道達到接收資料的目的，直到所需資訊完整接收完畢為止。

行動環境受限於廣播頻寬的不足，為服務廣大用戶端導致廣播主機負荷過重及資料壅塞，另外用戶端亦耗費大量接收設備的電力及時間來接收資料物件，所以為了提高無線廣播的效率，降低主機與用戶端耗費的成本，我們針對先前所介紹之廣播資料查詢環境，於本論文中進行「多重要求」及「非單調式」的特定廣播模式中接收效率之研究，主要探討無線廣播環境中「非單調式」重複資料物件，是否對於改善用戶端接收資料物件的時間有明確而顯著的幫助。在「單調式」及「非單調式」廣播循環環境中，對於用戶端效能量測是不同的，我們將於下節中進行描述。

## 第二節 單調式效能量測

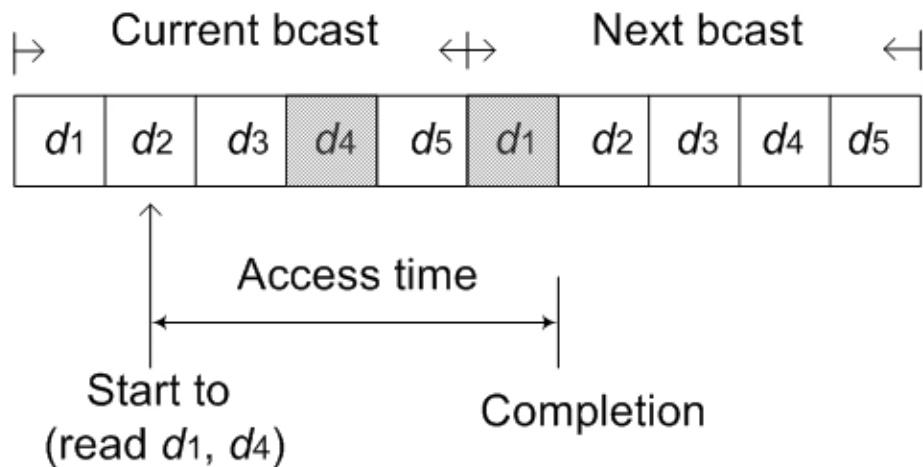
資料物件在廣播通道上的位置，關係用戶端能否快速地在短時間內擷取所需，無形中幫助降低了廣播主機的負擔以及用戶端所耗費的成本。無線廣播資料配置的問題，主要是找出一個廣播排程 $s$ 及其所對應之所有查詢的整體接收時間(Total Access Time; TAT)，TAT的值愈小愈好，也就是每一查詢的平均接收時間愈短愈好，我們以公式(1)說明：



$$TAT(s) = \sum_{q_i \in Q} AT^{avg}(q_i, s) \times freq(q_i) \quad \text{公式(1)}$$

$AT^{avg}(q_i, s)$ 是代表在資料廣播排程  $s$  中第  $i$  個查詢  $q_i$ ，所需的平均接收時間。圖四中說明了某一個用戶端在這個廣播環境中的接收時間，例子中一個查詢  $q_i$ ，需要接收兩個資料物件  $d_1$  及  $d_4$ ，開始接收資料物件如箭頭所標示的位置， $q_i$  進入時通道上時，正在廣播資料物件  $d_2$ ，而  $d_2$  並不是  $q_i$  所要的資料，所以需要等到  $d_3$  過後  $d_4$  才會出現；在同一週期中  $d_1$  被廣播是在  $q_i$  進入之前，所以要等到下一個廣播週期， $q_i$  才可能接收到  $d_1$ ，所以  $q_i$  所需的接收時間，如圖四中雙箭頭所示，因此  $q_i$  的平均接收時間跟  $q_i$  進來廣播頻道的開始點有很大的關係，要算平均接收時間是要考量每一個進來的點，故計算平均存取時間必須考量所有廣播頻道中可能進來的點，加總後取平均值，即是  $AT^{avg}(q_i, s)$ 。但是對於用戶端進入廣播頻道中接收資料物件的時機點是很難了解的，故對於  $AT^{avg}(q_i, s)$  的量化在現實生活中是有困難的。

想要找出一組好的廣播排程  $s$ ，使得  $TAT(s)$  愈低愈好，如前面所說用戶端平均接收時間  $AT^{avg}(q_i, s)$  是不好測量的；所幸在論文[11]中提出了一個新的且較容易計算的測量標準查詢距離(Query Distences;  $QD$ )，並且說明某個查詢  $q_i$  的  $AT^{avg}(q_i, s)$  跟  $q_i$  的查詢距離  $QD$  是成正比的；也就是對某一個  $q_i$  來說， $QD$  愈短則  $AT^{avg}(q_i, s)$  也愈短，兩者呈現正相關。

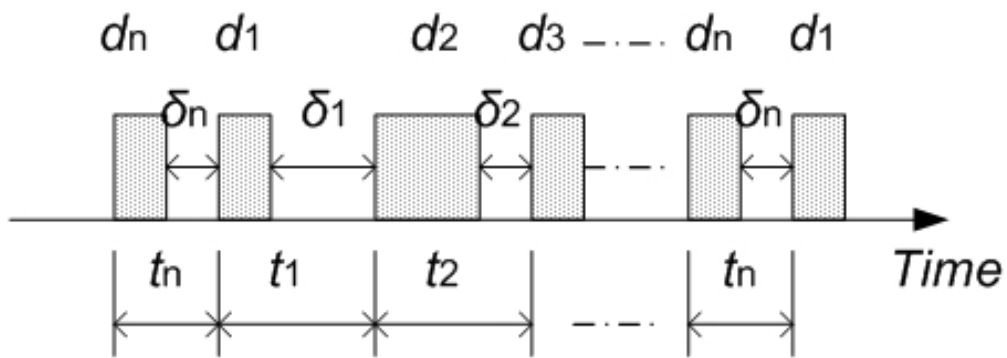


圖四 無線廣播中用戶端接收時間

以圖五為例，說明了在廣播頻道中某一個查詢 $q_i$ 所需的資料集將被廣播，假設  $QDS(q_i) = \{d_1, d_2, \dots, d_n\}$ ， $n$  為  $q_i$  所要接收資料物件的數量， $d_i$  表示資料物件  $d_j$  及  $d_{j+1}$  兩者間在廣播順序中的距離，就是  $t_j - |d_j|$ ， $t_j$  就是  $d_j$  起始到下一個  $d_{j+1}$  開始這中間的長度，因為資料物件採循環式廣播，就是每隔一段時間就會重覆廣播，所以所有  $t_j$  的總合就是一個 bcast 的大小， $q_i$  的查詢距離如公式 (2)：

$$QD(q_i, s) = B - \text{MAX}(d_k), k = 1 \dots n. \quad \text{公式(2)}$$

舉例來說，如圖四所示，若以  $QD$  測量的方法來說明，當  $QD(q_i) = \{d_1, d_4\}$ ， $d_1$  所指的是  $d_1$  及  $d_4$  間的距離，中間隔了  $d_2$  及  $d_3$ ，故  $d_1 = 2$ ， $d_2$  所指的



圖五 查詢資料集配置

是  $d_4$   $d_1$  的距離，中間隔了  $d_5$ ， $d_2 = 1$ ，因此導入公式  $QD(q_i, s) = B - \text{MAX}(d_k)$ ,  $k = 1 \dots n$ ，長度  $B = 5$ ， $\text{MAX}(d_k) = 2$ ，故  $QD(q_i, s) = 5 - 2 = 3$ 。

我們假設的前提是每一個資料物件長度都相等，無線廣播的問題是要找出一個好的廣播排程，讓  $QD$  總長度愈短愈好，我們從前段得知  $TAT$  越短則  $q_i$  的平均接收時間也就愈短，也就是說對某一個  $q_i$  來說  $AT^{avg}(q_i, s)$  跟  $QD$  呈現正比關係，因此目的就是求得  $SQD(q_i, s) \times \text{freq}(q_i)$  之最小化。本論文有關單調式查詢距離量測所用原始程式碼收錄於附錄二中，距離量測函數的設計與問題大小呈現線性關係，故其時間複雜度為  $O(n)$ 。

### 第三節 非單調式效能量測

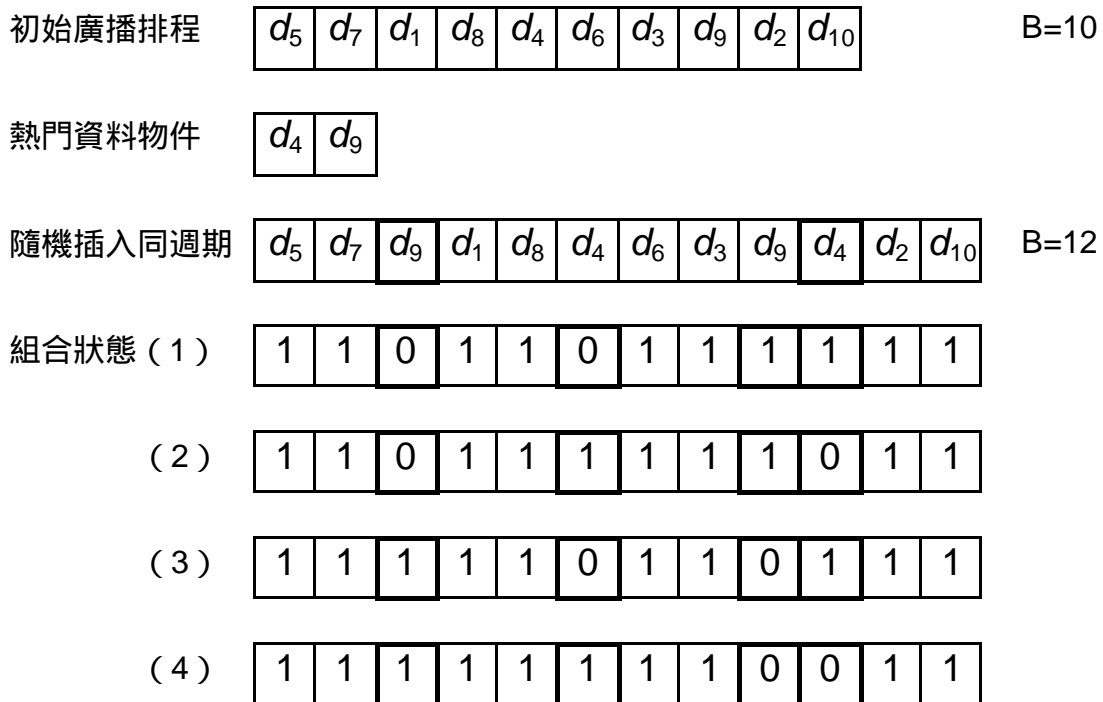
上述對於無線廣播的環境僅限於在同一廣播週期中，廣播資料物件是唯一一個，也就是不可重複，屬於單調式廣播方式，其有關查詢距離  $QD$  的量測爰依論文[11]中，所提出一種較新也較容易量化與計算的測量

標準。

本論文中統計熱門資料後，並且將熱門的資料物件，隨機方式插入同一廣播週期中，形成同一廣播週期中，有數個資料物件的重複，用來幫助我們了解，為大多數人所期望接收的物件資料，當重複出現時，對於降低用戶端的平均接收時間的程度。因此對於一個用戶端的需求而言，在同一週期中假定有二個資料物件重複，量測標準如依照前揭所載方式，廣播序列的排列組合就會出現四種組合情形，假定有三個資料物件重複，廣播序列的排列組合就會有八種情形，這對於廣播主機的排程演算是相當的耗費時間，其時間複雜度則呈現最壞狀況(Worst-Case)  $O(2^h)$ ， $h$  表示同一廣播週期中重複的資料物件數，我們以圖六為例。

初始廣播資料物件序列，同一週期中資料物件是不重複的， $s = \langle d_5, d_7, d_1, d_8, d_4, d_6, d_3, d_9, d_2, d_{10} \rangle$ ，但假定每個資料物件經過統計分析後，熱門程度呈現， $f(d_4) > f(d_9) > f(d_1) > f(d_7) > f(d_{10}) > f(d_2) > f(d_5) > f(d_8) > f(d_6) > f(d_3)$  的情形，假設我們取前兩個最熱門的資料物件  $D_h = \{d_4, d_9\}$ ，以隨機方式插入廣播序列中，形成同一週期中存在重複的資料物件， $s = \langle d_5, d_7, d_9, d_1, d_8, d_4, d_6, d_3, d_9, d_4, d_2, d_{10} \rangle$ ，如爰用上節公式(2)，來對每一個用戶端的需求做距離量測，廣播序列具有  $2^h$  種組合方式，在此有兩個重複資料物件，故  $h = 2$ ， $2^h = 4$ ，代表會有四種組合情形，如此每一個用戶端就需要做  $2^h$  次的量測距離演算，也就是每一個用戶端由原本的量測一次，增加成為  $2^h$  倍率，其時間

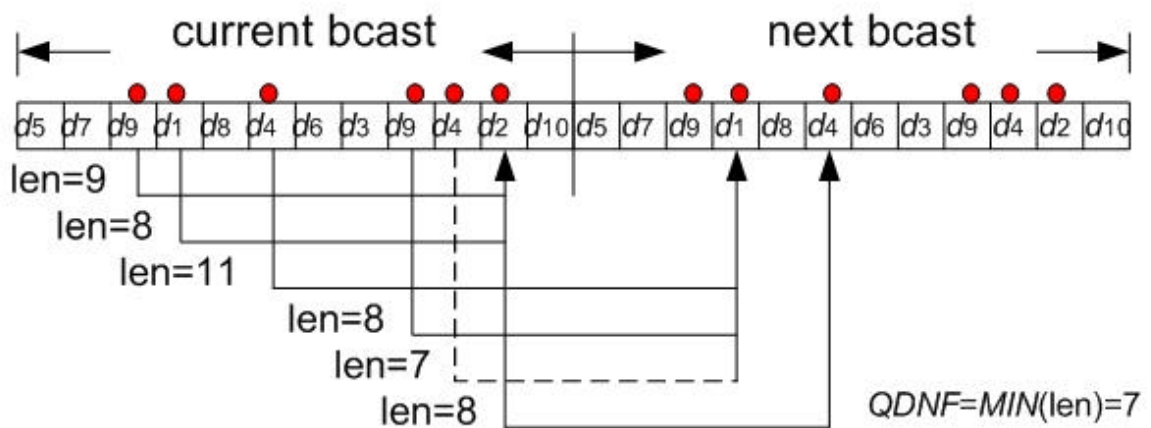
複雜度則處在一種極差狀況  $O(2^h \times n)$ ，這對於廣播主機的負擔無疑是雪上加霜。



圖六 廣播週期中重複資料物件與距離量測關係圖

當我們嘗試了解利用合宜的熱門資料物件數量，插入單一週期的廣播序列中，對於降低使用者查詢資料時間是否有幫助以及可能出現問題的同時，發現了如前段所述的問題，故上一節公式(2)所提有關距離量測方法並不適用在非單調式廣播環境中，為達到本論文目的，並且能夠順利降低廣播主機對於排程演算的時間複雜度，我們提出了另一種廣播環境中有關非單調式距離測量(Query Distences of Non-Flat; QDNF)的方法，以圖七為例說明。

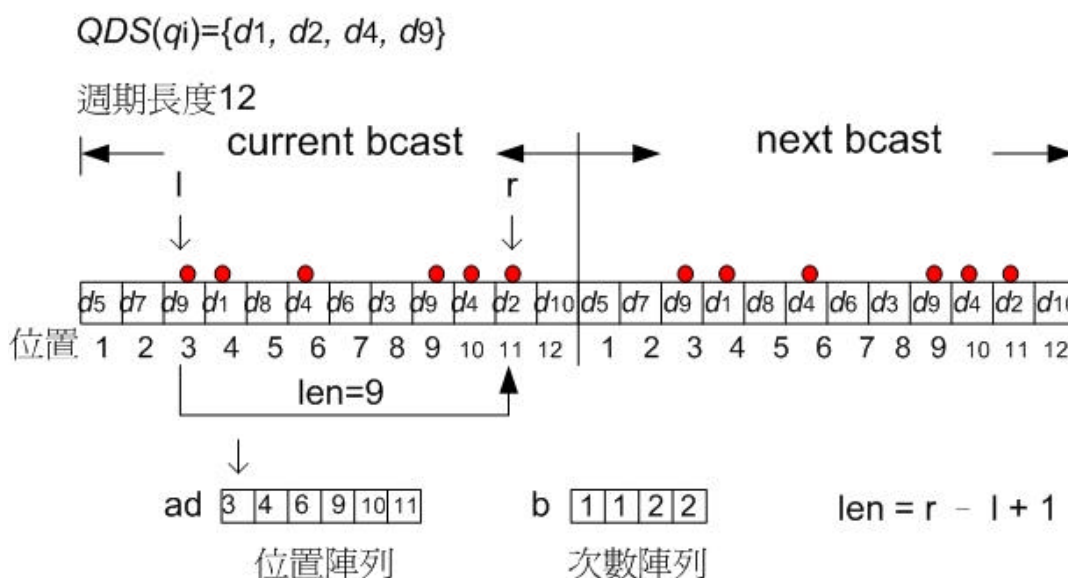
圖七為一組內有兩個，以隨機方式插入同一廣播週期中的熱門資料物件  $d_4$  及  $d_9$ ，假定一用戶端的需求呈現  $QDS(q_i)=\{d_1, d_2, d_4, d_9\}$ ，圖中圓圈處代表週期中，符合  $q_i$  所需資料物件的位置，從這些位置起，直到滿足  $q_i$  所需資料物件的距離，就如圖上標示線條處，在同一週期中， $q_i$  需求四個資料物件，其中兩個為被重複的熱門資料物件，所以在同一週期中，共有六個能滿足  $q_i$  需求的位置，在演算的過程中，我們利用掌握這六個位置，依序由這六個位置往後找到滿足  $q_i$  需求的最短距離  $QDNF=MIN(len)=7$ 。



圖七 週期中重複資料物件距離量測圖

我們以現行週期第一個被需求的資料物件  $d_4$  為例，從現行週期  $d_4$  這位置起，用戶端無法在同一週期中完全取得  $d_1$ 、 $d_2$ 、 $d_4$  及  $d_9$ ，是必須跨越至下一廣播週期中，直到  $d_1$  這個位置， $q_i$  需求才得以滿足，故從開始在廣播頻道上傾聽直到完成，所需距離長度為 11。

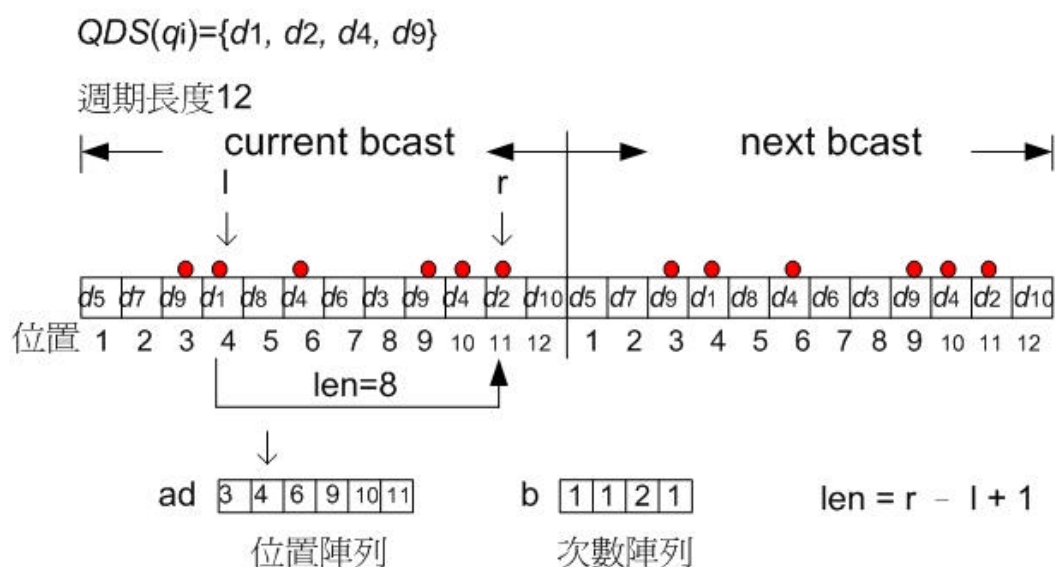
在現行週期中這六個位置，導出個別的距離長度分別如圖七所標示，其中最短的距離為圖上虛線部分，距離為 7。這樣的演算法相較於前述，能夠讓原先效率頗差的時間複雜度在  $O(2^h \times n)$  狀態下，降低至線性時間  $O(n)$ 。至於如何能在滿足  $q_i$  需求的六個位置上逐一計算出最短距離長度，我們將藉由下段詳細說明演算的方法。



圖八 週期中重複資料物件距離量測圖

圖八我們仍然延續前例，我們設計了兩個陣列，陣列 ad 主要紀錄  $q_i$  所需的資料物件出現在廣播週期中的位置，分別是在廣播週期中第 3、4、6、9、10 及 11 的位置，陣列 b 主要是相對紀錄  $q_i$  所需的資料物件出現在廣播週期中出現的次數，第一個資料物件  $d_1$  出現在廣播週期中的次

數為 1 次， $d_2$  為 1 次， $d_4$  為 2 次， $d_9$  為 2 次，接下來我們在現行週期中找到前後兩個為  $q_i$  所需的資料物件位置指標  $l$  及  $r$ ， $l$  表示  $q_i$  進入週期中開始接收資料物件的時間點，我們利用陣列  $ad$  第一個位置內容 3，指向週期中第 3 個位置， $r$  則代表  $q_i$  已經滿足需求並且結束接收的時間點，次數陣列  $b = [1, 1, 2, 2]$  中顯示出，開始接收  $l$  到結束接收  $r$  中間，所需要的資料物件均已經出現（陣列  $b$  中沒有任何一個數字為 0），量測距離長度為  $r - l + 1$ ，故距離為 9。

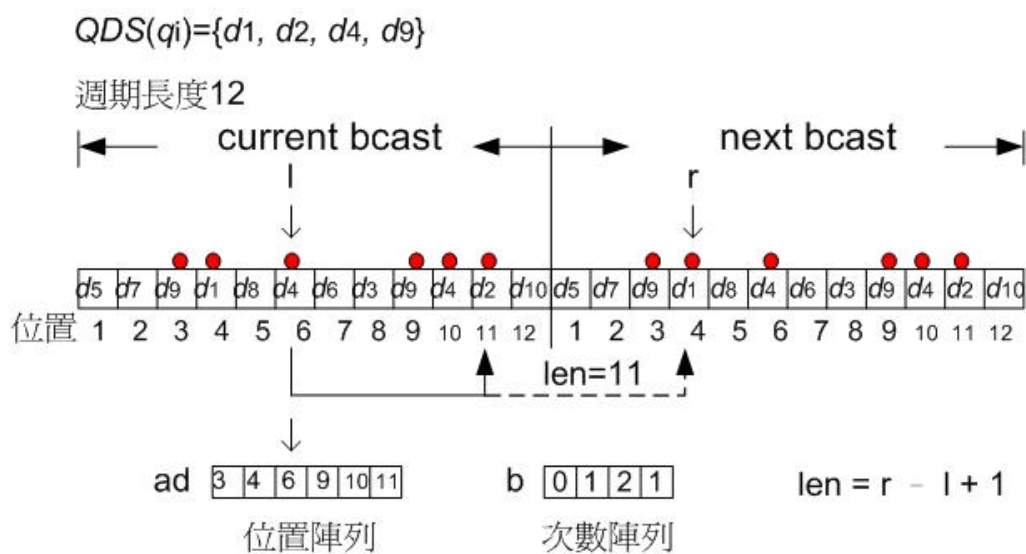


圖九 週期中重複資料物件距離量測圖

圖九位置陣列  $ad$  第二個位置內容 4，指向第二個  $q_i$  所需資料物件出現在廣播週期中的位置 4， $l$  指標由原先廣播週期第 3 個位置移動到第 4



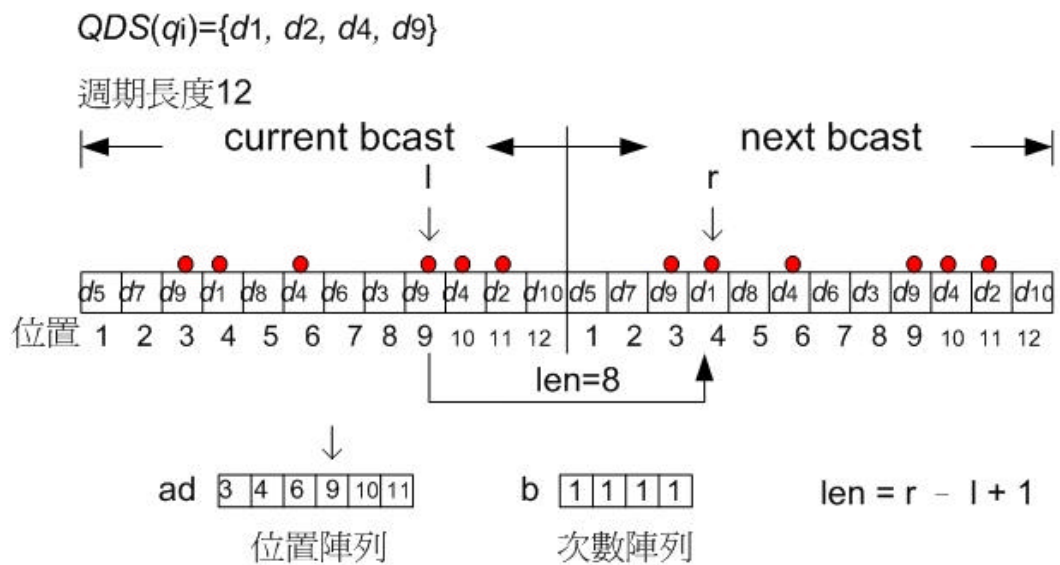
個位置上，表示  $l$  及  $r$  中間， $d_9$  出現的次數少了一次，故次數陣列  $b$  中，相對於  $d_9$  位置的內容則需要減 1，形成次數陣列  $b = [1, 1, 2, 1]$ ，但仍表示從開始接收  $l$  到結束接收  $r$  中間，所需要的資料物件均已經出現，量測距離長度同樣為  $r - l + 1$ ，故距離為 8。



圖十 週期中重複資料物件距離量測圖

圖十位置陣列  $ad$  第三個位置內容 6，指向第三個  $q_i$  所需資料物件出現在廣播週期中的位置 6， $l$  指標由原先廣播週期第 4 個位置移動到第 6 個位置，表示  $l$  及  $r$  中間， $d_1$  出現的次數少了一次，故次數陣列  $b$  中，相對於  $d_1$  位置的內容則需要減 1，形成次數陣列  $b = [0, 1, 2, 1]$ ，這時候表示從開始接收  $l$  到結束接收  $r$  中間，少了一個所需要的資料物件  $d_1$ ，為了

滿足  $q_i$  需求， $r$  指標則需要往後找到  $d_1$ ，圖十中可以看出指標  $r$ ，無法在現行週期中找到  $d_1$ ，必須跨越到下一週期第 4 個位置才找到  $d_1$ ， $q_i$  才得以滿足，量測距離長度同樣為  $r - l + 1$ ，故距離為 11。



圖十一 週期中重複資料物件距離量測圖

圖十一，前述  $r$  指標跨越到下一週期中，找到資料物件  $d_1$ ，滿足了  $q_i$  需求，次數陣列中相對於  $d_1$  位置的內容則需要加 1。位置陣列第 4 個位置內容 9，指向第四個  $q_i$  所需資料物件出現在廣播週期中的位置 9， $l$  指標由原先廣播週期第 6 個位置移動到第 9 個位置，表示  $l$  及  $r$  中間， $d_4$  出現的次數少了一次，故次數陣列  $b$  中，相對於  $d_4$  位置的內容則需要減 1，故次數陣列  $b = [1, 1, 1, 1, 1]$ ，這表示從開始接收  $l$  到結束接收  $r$  中間，

所需要的資料物件至少都出現一次，量測距離長度同樣為  $r - l + 1$ ，故距離為 8。由上可知藉著掌握廣播週期中滿足用戶端需求資料物件的位置及次數，我們可以很快的計算出最短的距離，意即  $QDNF$  的值，這讓原本呈現不佳的時間複雜度，下降回歸到線性時間  $O(n)$ 。

在非單調式的廣播環境下，透過我們所設計的演算方法，找到一組最佳的廣播排程，我們當然期望其所對應的所有用戶端需求的總接收時間  $TAT$  值越小越好， $TAT()$  公式如下：

$$TAT() = \sum_{q_i \in Q} AT^{avg}(q_i) \times freq(q_i) \quad \text{公式(3)}$$

對於某查詢  $q_i$  的平均接收時間越短，則相對  $TAT$  時間就會越短，我們於本節中提出了一種對於用戶端接收時間的量測方法，對  $q_i$  來說同樣的  $AT^{avg}(q_i)$  與  $QDNF$  呈現正比關係，因此我們的目的同樣是求得公式(4)之最小化。本節所述非單調式查詢距離量測原始程式碼收錄於附錄三中，供參考指正。

$$TAT() = \sum_{q_i \in Q} QDNF(q_i) \times freq(q_i) \quad \text{公式(4)}$$

## 第三章 我們的演算法

在研究中，我們必須先行統計了解資料物件相對於用戶端的需求，也就是將個別資料物件依熱門程度排列。個別資料物件就是因為具有不同程度的被需求性，將熱門資料物件插入單一週期的廣播序列中，用來快速滿足多數用戶端對該物件需求，接續藉由模擬退火演算法模擬金屬退火過程，進行廣播排程最佳化的處理，用以得到最為有效的廣播序列。最後經由實驗了解得到合宜的熱門資料物件數量插入單一週期的廣播序列中，對於降低使用者查詢資料的時間是非常有幫助的。以下我們將詳盡說明整個方法及過程。

### 第一節 熱門資料物件的處理

當用戶端將其需求傳送至廣播主機時，每個資料物件的被需求次數有一定程度上的差異，因此廣播主機依照用戶端的需求或者歷史資料進行統計，便於了解各個資料物件的熱門程度，公式(5)說明了一個資料物件它所受歡迎的程度：

$$f(d_i) = \sum_{d_i \in QDS(q_j)} \text{freq}(q_j), 1 \leq j \leq |Q| \dots\dots \text{公式(5)}$$

由公式(5)中我們可以了解個別資料物件熱門程度，主要來自各個資料物件被每一個用戶端所需求頻率的總和。廣播主機在單一週期中以隨機方

式產生一組不重複資料物件的廣播初始序列，我們嘗試將熱門程度較高的資料物件，以隨機方式插入此一廣播週期中，使的單一廣播週期長度變長，就是因為插入的資料物件熱門程度高，為多數用戶端所需要，故對於用戶端而言，單一廣播週期中有了重複的資料物件，便可以提供用戶端更快速的取得所需。

我們假定以六組用戶端 $q_i$ 所需求資料物件集合為例，分別為  
 $QDS(q_1)=\{d_1, d_4, d_9\}$ 、 $QDS(q_2)=\{d_2, d_4, d_5, d_6, d_8\}$ 、 $QDS(q_3)=\{d_1, d_4, d_7, d_9\}$ 、 $QDS(q_4)=\{d_3, d_4, d_7, d_9\}$ 、 $QDS(q_5)=\{d_1, d_4, d_7, d_9, d_{10}\}$ 以及 $QD(q_6)=\{d_4, d_7, d_9, d_{10}\}$  各組的查詢頻率分別為 $freq(q_1)=9$   $freq(q_2)=5$   $freq(q_3)=7$ 、 $freq(q_4)=3$ 、 $freq(q_5)=8$ 及 $freq(q_6)=2$ 。當查詢資料上傳至主機後，廣播主機採聯集 $D = \cup q_i$ ，得到一組廣播資料物件 $D=\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}\}$ 的集合，並且將各個資料物件依照公式(5)加以計算，例如 $d_1$ 它的熱門程度計算如下：

$$\begin{aligned} f(d_1) &= freq(q_1) + freq(q_3) + freq(q_5) \\ &= 9 + 7 + 8 = 24 \end{aligned}$$

我們將廣播資料物件集合 $D$ 的個別資料物件，經熱門程度計算公式後，整理如表二。

資料物件	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$	$d_9$	$d_{10}$
$f(d_i)$	24	5	3	34	5	5	20	5	29	10

表二 廣播資料物件熱門程度統計表

由上表可知資料物件 $d_4$ 為大多數用戶端所需求的資料物件，也就是相當熱門的資料物件，相對於 $d_3$ ，資料物件 $d_3$ 就顯得格外冷門，然而就現實環境中，的確存在一些為大多數人所喜愛偏好的資料，例如當紅股票上市公司股價或汽車上所使用的衛星導航生活資訊的提供等，伺服器主機可以透過不同管道取得資訊，將這些用戶端需求或者歷史資料加以整理統計後，分析出為多數人所喜愛的資料物件。

## 第二節 廣播資料物件的處理

上節中廣播主機依照各個資料物件做了熱門程度的分析統計後，但是主機仍然需要知道，那些資料物件將要拿來放進廣播週期中，故我們便將各個資料物件熱門程度分析後的結果，再利用任何一種排序方法加以排序，也就是將資料物件由熱門往冷門程度排列，如表三。有關資料物件熱門程度的統計分析原始程式碼收錄於附錄一中，請參閱。

資料物件	$d_4$	$d_9$	$d_1$	$d_7$	$d_{10}$	$d_2$	$d_5$	$d_8$	$d_6$	$d_3$
$f(d_i)$	34	29	24	20	10	5	5	5	5	3

表三 排序後熱門資料物件表

一組好的廣播序列對於降低用戶端的接收資料物件時間是非常有幫助的，但廣播主機並不了解廣播資料物件做怎樣的排列組合，是有助於幫助用戶端的，故廣播主機於初始，將所有被查詢不重複之資料物件

於單一週期中進行隨機式的廣播順序配置，如圖六所示。

由於在單一週期中資料物件是不重複的，爰依公式(2)中所定義運用查詢資料間隔距離，來進行廣播資料物件序列相對用戶端的距離量測。由於初始的廣播序列對於實驗結果，具有一定程度的影響，為使各次實驗均能處於同一基準點之上，故本論文將初始隨機所產生不重複的廣播序列寫入檔案，以利各次實驗所引用。

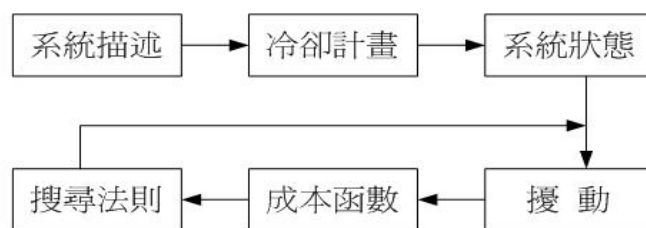
有了初始廣播資料物件的序列後，我們假定越是熱門的資料物件如能放入初始序列中，對於滿足多數用戶端需求，降低用戶端平均接收時間應該是很有幫助的，基於這樣的一個想法，我們每次選取熱門程度最高的資料物件，隨機插入初始廣播序列中，插入的熱門資料物件數量就是同一廣播週期中重複的資料物件數  $h$ ，至於每次實驗插入的熱門資料物件數量，依公式  $h=2 \times r$ ， $r$  為次數，例如第一次  $r=1$  時，所需要插入的熱門資料物件數量  $h=2 \times 1=2$ ，就是取前兩個最熱門的資料物件  $D_h=\{d_4, d_9\}$  插入初始序列中，當第二次時，熱門資料物件數  $h=4$ ，也就是需要  $D_h=\{d_4, d_9, d_1, d_7\}$  等四個資料物件來插入初始序列中。每次實驗均需透過模擬退火演算法，幫助找到最佳的廣播序列排程，由於插入一定數量的熱門資料物件後，形成同一廣播週期中出現了一定數量的重複資料物件，在此非單調式的廣播環境中，對於用戶端所進行距離的量測是不同於單調式的，我們仍依照第二章第三節所載方式進行量測。

隨著插入的熱門資料物件數量越來越多，一個廣播週期的長度也隨

著越來越長，或許我們會陷入於越多資料物件的插入，對於降低用戶端接收時間越有幫助的迷思當中。我們當嘗試了解，對於一定數量的資料物件預備插入初始序列之中時，並不是每個熱門資料物件都是非常熱門的，先前我們有敘述過每個資料物件都有其受歡迎的程度，這些即將插入的熱門資料物件或許絕大部分均非常受歡迎，但部分有可能屬於較為不熱門或者是不熱門的資料物件，就因為有這些較不受歡迎物件的插入，進而影響用戶端平均接收的時間，我們於實驗中將會以數據方式呈現其狀態。

### 第三節 模擬退火法

Kirkpatrick等人在1983年提出模擬退火法[18]，並成功地應用在解決最佳化的問題，它是蒙地卡羅演算法的推廣。模擬退火法不同於鄰近搜尋法，具有跳出區域最佳解，而達到全域最佳解的能力。以下將透過圖十二所示，逐一簡要說明本研究運用模擬退火法的設計及運算方法，應用於無線廣播求取最佳廣播排程的過程。



圖十二 模擬退火法描述概圖



系統描述(the description of the system) :

在無線廣播的領域中， $s = d_1, d_2, d_3, \dots, d_n$ ，代表一組廣播的序列，其中 $d$ 代表一個即將被廣播的資料物件，資料物件相當於金屬結構分子，加熱到一定溫度後，所有分子在狀態空間中自由運動，隨著溫度下降，這些分子逐漸停留在不同狀態，而分子的狀態分布則滿足波茲曼 ( Boltzmann ) 概率分布函數。

冷卻計畫(Cooling Schedule) :

將任何可能近似最佳廣播排程，由紊亂之隨機狀態逐漸導引至穩定狀態的降溫過程。期間所須決定的參數包括：

A. 起始溫度(initial temperature) :  $T_i$

起始溫度必須夠高，高到足以被所有的解所接受，並具有可以離開或不會侷限於區域最佳解的能力，其設定並無絕對標準或大小，仍需視研究問題的規模而定。

B. 溫度長度(temperature length) :  $k$

$k$ 即馬可夫鏈的長度，指在同一溫度  $T_i$  下，將反覆進行擾動  $k$  次，也就是當達到一個溫度長度時，系統才會進行降溫動作。此一長度與問題規模有關，如 Bonomi 和 Lutton (1984) 設定為問題大小乘以某倍數等。但是馬可夫鏈的長度仍必須受到限制，避免求產生一個不合理的求解時間。

C. 冷卻率(cooling ratio) :  $a$

冷卻率是指降溫時機已屆時，由現行溫度下降到次一溫度的比率。一般的降溫方式為  $T_i = aT_{i-1}$ ， $a$  係數的值介於 0.8~0.99 之間。

D. 凍結條件(frozen)：  $T_f$

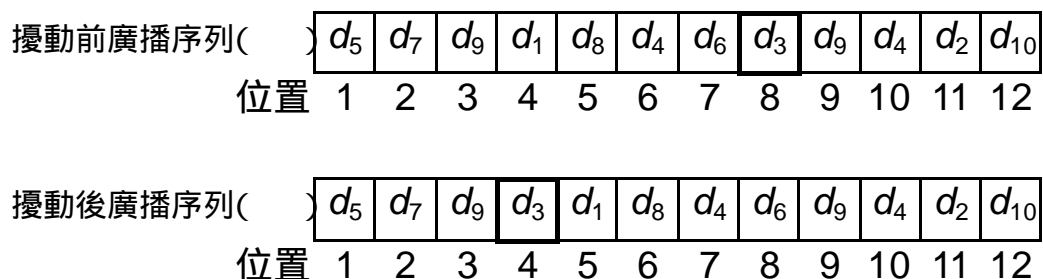
使模擬退火法演算終止的法則。凍結條件最單純的設定方式，是指定一個固定的終止溫度，一般是一個趨近於零的極小值，本文凍結條件設為  $T_f < 1$ ，意即溫度逐次下降至小於 1，演算程序即將終止。另還有其他檢查求得解是否有所改善的方式，如 Kouvelis and Chiang(1992) 設定若經數次降溫後，求得的最佳解仍未改善或者可行解遞移接受的比率低於一定值，則終止運算。

系統狀態(Configuration)：

廣播資料物件序列所有可能的組合。

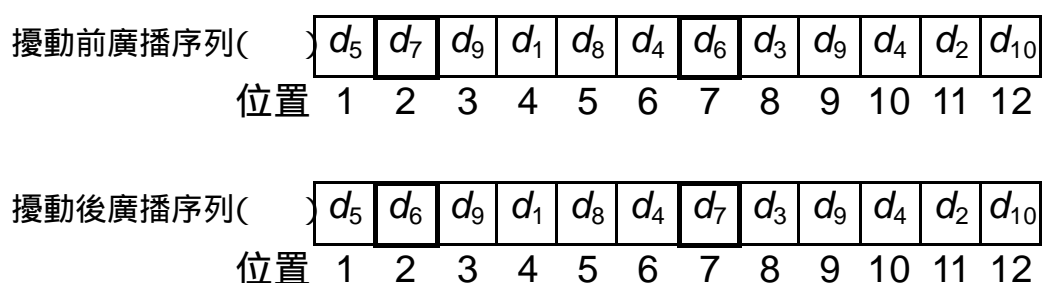
擾動的方法(the perturbation mechanism)：

擾動的目的是為了讓原有的廣播序列產生不同的變化，用來找到更好的解。本研究在每次的擾動中，使用隨機方式選取兩種擾動方法的其中一種，第一種擾動方法 ( op1 ) 為隨機選取一資料物件插入廣播序列中



圖十三 擾動方法 ( op1 )

的任一位置，例如圖十三廣播序列，隨機選取第8個位置的資料物件  $d_3$ ，並且將該物件隨機插入至第4個位置，原先第4個位置起到第7個位置的資料物件全部往後移動一個位置。第二種擾動方法 (op2) 為隨機選取廣播序列中兩資料物件相互對調，例如圖十四，隨機選取第2及第7個位置的資料物件  $d_7$  及  $d_6$  進行對調。擾動方法由於是透過演算法隨機選取，兩種方法被選取的機率約各佔50%。



圖十四 擾動方法 (op2)

成本函數(Cost Function)：

衡量廣播序列經由擾動後，對於用戶端的品質，在此主要還是在量測擾動後產生一組新的廣播排程，其對於用戶端的總接收時間  $TAT$ ，就如同公式(4)，我們使用  $QDNF$  的量測方法進行量測，並以  $Cost()$  對應到公式(4)中，相關量測標準與方法如本文第二章所述。

搜尋法則(Move Set)：

退火過程中，由一個廣播排程移動到另一個廣播排程的規則，即將目前的可行解遞移(Transit)至另一可行解。前一組廣播序列  $s_i$ ，其經由成

本函數計算得到 $C_i$ ，經由擾動後得到一組新的序列 $i_{+1}$ ，同樣經由成本函數計算得到 $C_{i+1}$ ， $C = C_i - C_{i+1}$ ，假設 $C > 0$ ，則進行可行解遞移， $i = i_{+1}$ ， $C_i = C_{i+1}$ ，若 $C < 0$ ，則產生一個uniform(0,1)的亂數 $R$ ，並計算 $P = \exp(-C/T_i)$ 。若 $p > R$ ，則令 $i = i_{+1}$ ， $C_i = C_{i+1}$ 。若 $p < R$ ，則 $i$ 仍為目前的最佳解。模擬退火法演算法如下：

```

Star with a random solution  $i$ 
 $C_i = \text{Cost}(i)$ 
While  $T_i > T_f$ 
    Loop  $k$ 
        Using op1 or op2 ( Produces a neighbor solution  $i_{+1}$ )
         $C_{i+1} = \text{Cost}(i_{+1})$ 
         $C = C_i - C_{i+1}$ 
        If  $C > 0$  Accept  $i_{+1}$ 
        else
             $R = \text{uniform}(0,1)$ 
             $P = \exp(-C/T_i)$ 
            If  $P > R$  Accept  $i_{+1}$ 
            else
                reject  $i_{+1}$ 
        EndLoop
         $T_i = aT_i$ 
    EndWhile

```

本研究將初始的一組廣播序列，在退火過程中不斷擾動及計算成本函數，試圖找到一組較好的廣播序列，好的序列當然需要保留，但是為了在過程中能擁有離開區域最佳解的機會，較差的序列則交由波茲曼概率來決定，用以確保能夠往整體最佳解來前進。

## 第四章 實驗結果

### 第一節 實驗環境

為求得一個問題的結果，我們通常需要透過實驗的過程，但是實驗室的環境與實際環境中間顯然存在一些差異，當然我們期望獲得的結果是科學與接近事實的，所以當我們進行本研究探討與實驗時，我們儘可能的將問題單純化，並且將相關條件定義清楚，以利我們對於實驗結果的研究。

有關本研究探討行動環境中可重複資料物件的廣播問題，就實驗所用測試設備其配備 Intel Pentium 2.4G的CPU與主記憶體RAM為512MB，使用Visual C++來撰寫程式，並且在微軟視窗作業系統Win 98平台下做測試。實驗中我們並不特別關心中央處理器在整個實驗過程中所扮演的角色，因科技進步一日千里，我們強調的是廣播主機在做排程演算時，所花費的時間應該是合理且可以容忍的，當然要能夠達到上述目的，其因素不僅僅是硬體的條件，另一方面所設計的演算法亦相當關鍵。

在此我們對於模擬的廣播環境做一些清楚的定義，用戶端可透過上傳需求至伺服器端，所送出要求的資訊可以包含兩個以上的資料物件，而這些資料物件的大小是固定的，廣播伺服器端接收到這些「多重需求」後，會放置在伺服器之中，伺服器根據所有用戶端的需求進行統計分析，找到每一個資料物件的熱門程度後，將熱門

資料物件插入初始廣播序列中，並且在合理時間範圍內，經由廣播排程演算，找到一組最佳的廣播序列，然後將資料物件依序送往廣播頻道上進行廣播，取代現行廣播排程之內容。用戶端則可以透過監聽廣播頻道，當自己所需要的資料物件到達時，即透過下載頻道接收資料物件，直到用戶端所有需求的資訊都被滿足。我們假設使用者端皆可儲存任何在廣播頻道中所需求的資料，並且所有資料皆可於廣播頻道中獲得，因此降低用戶端平均接收時間，則是我們所要研究的方向。在假設的環境中，我們簡化所有的環境因素，並聚焦於合宜的重複資料物件數，並以模擬退火法協助我們找到一組最適的廣播序列。我們將就逐漸增加熱門資料數量，了解是否能幫助用戶端降低平均接收時間進行效能的評估。

## 第二節 實驗資料及過程

開始實驗之前，首先我們必須利用程式碼造出實驗所必須的資料檔，資料檔最主要用來模擬，用戶端對於資料物件需求變化的情形。依照查詢頻率分佈狀況情形，我們造就了常態分布 (normal) 以及指數分布 (exponential) 的資料庫，來模擬現實生活中用戶端的查詢狀況。

實驗環境中，使用常態分布資料檔主要是模擬廣播主機所廣播的資料物件，有一些是一般用戶端所感興趣的，譬如每日新聞頭條或是股價財經資訊的提供；而另一些資料物件則是用戶端比較不感

興趣的資料，符合一般用戶端查詢的習慣。另外指數分布資料檔，主要是在呈現現實生活中，有些資料是當下極端為用戶端所需要的，例如造成全國性的話題新聞或是足以撼動人心吸引絕大多數人目光

程式名稱	statistice.cpp	bdlrme_single.cpp	bdlrme_replication.cpp
讀入檔案	ed500q200s002.txt	ed500q200s002.txt	ed500q200s002.txt nd500q200s002.txt
	nd500q200s002.txt	nd500q200s002.txt	inzi_sch.txt popular_data.txt
輸出檔案	popular_data.txt	inzi_sch.txt	result_file.txt
		result_file.txt	
說明	<p>ed500q200s002.txt及nd500q200s002.txt為實驗用指數型及常態型資料檔，具有500資料物件數、200用戶端及每一用戶端有10資料物件需求數。</p> <p>statistice.cpp依據資料檔的內容，統計熱門資料物件，並且加以排序，並將結果寫入popular_data.txt。</p> <p>bdlrme_single.cpp產生不重複資料物件的初始序列，並記錄於inzi_sch.txt，經模擬退火法演算之後，將結果紀錄於result_file.txt。</p> <p>bdlrme_replication.cpp於初始序列中，逐次以一定數量的熱門資料物件，以亂數方式插入其中後，經模擬退火法演算，將結果紀錄於result_file.txt。</p>		

表四 實驗程式與檔案輸出入情形



焦點的事件，相對於大多數人因過度轉移聚焦而忽略一般會去關心的議題，這也就是指數型資料型態的特性。而另外在實驗中我們忽略了均勻型（uniform）資料檔，均勻型資料檔呈現所有資料物件被需求的頻率都是一致的，這與現實生活中資訊被需求的情形明顯有差異，故本研究並不採用均勻型資料檔。

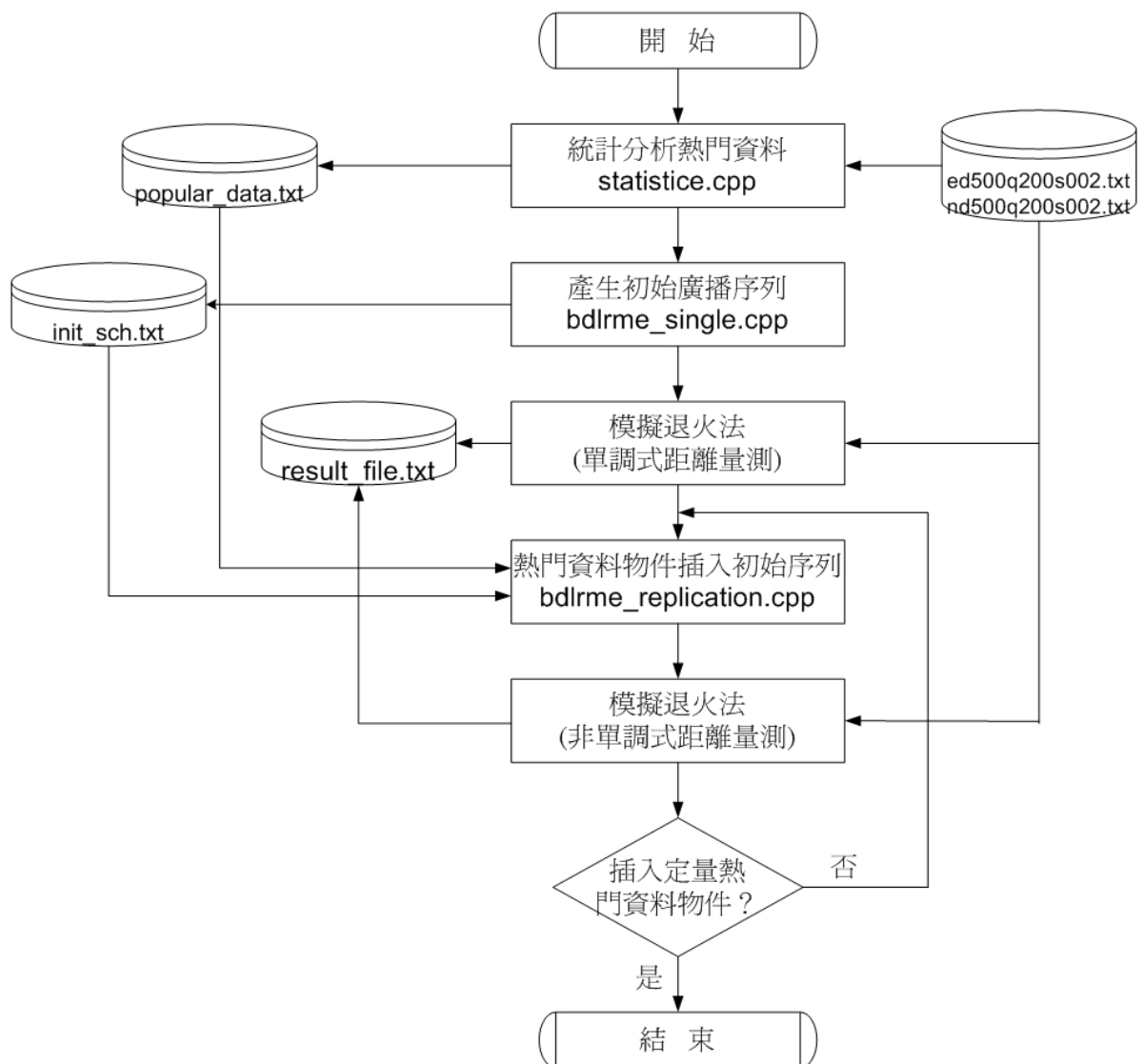
製作實驗用資料檔，除了依查詢頻率分布為其參考因素之外，我們假定每一個資料檔各具有500資料物件數、200用戶端及每一用戶端有10資料物件需求數。

有了資料檔，對於每個資料物件熱門程度我們做了統計之後，便將熱門資料物件逐次隨機插入初始廣播序列之中，經由模擬退火法不斷擾動、計算距離、降溫等過程的演算，找到最終的一組廣播序列，此序列或許不是最佳序列，但是起碼是一組在合理時間範圍內，為多數用戶端可以接受的廣播序列。

論文中有關模擬退火法降溫過程所使用的相關參數我們設計如下，起始溫度  $T_i = 1000$ 、溫度長度  $k$  在本研究中採用資料物件數量乘以20為計算基礎（ $k = 500 \times 20 = 10000$ 次）、冷卻率  $a = 0.95$  以及凍結條件  $T_f < 1$ ，實驗過程中我們並且紀錄了查詢距離變化的情形。

表四描述了整個實驗過程中，所使用的程式檔及資料檔案進出的情形，主要區分統計熱門資料物件、廣播週期中產生不重複的廣播排程及廣播周期中開始插入熱門資料物件等三個部分。其中廣播序列重複或者

不重複，均會透過模擬退火法來做最佳廣播排程的演算，但有關總距離的計算，所引用的演算方法卻是不同，這在本論文第三章有關問題描述中已載明，請參閱。



圖十五 實驗過程及步驟

圖十五說明了整個實驗過程及其步驟，在此我們除了了解插入熱門資料物件對於用戶端降低平均接收時間的幫助之外，我們也期望了解到底需要插入多少熱門資料物件是合理的，過與不及都可能產生一些負面效果，熱門資料插入少對於整體效能並不會明顯，另外太多的熱門及不熱門的資料物件插入序列中，反而造成整體效能下降，在圖十五中最後一個決策圖形，主要就是在描述上述的關係，幫助我們了解適當的插入資料物件數量重要性。

### 第三節 結果分析

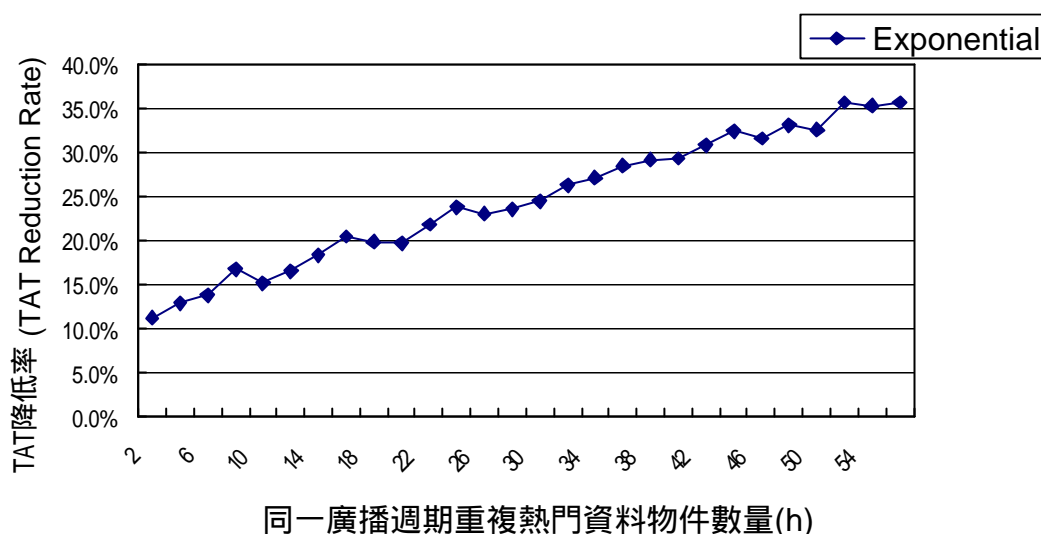
實驗中假設一個查詢需要多個資料物件、單一個廣播週期中資料物件是可以重複的以及資料物件無順序性，我們分析整個實驗效能，主要分成兩個部分，第一個部分是觀察廣播序列中未重複資料物件與開始插入熱門資料物件後，同樣經由模擬退火演算法所得到的總距離量測間差距變化的情形，此部份可以幫助我們去了解，週期中重複熱門資料物件對於用戶端所產生的影響以及可能面臨的問題。第二部份我們期望了解類比化模擬退火法於解廣播序列最佳化的問題上所獲致的成效如何，並觀察廣播序列未經及經過了模擬退火演算法之後，總距離量測間差距變化的情形，即  $TAT$  降低率 (Reduction Rate) 變化情形， $TAT$  降低率數值越大則呈現越好的狀態，有關的參數敘述如下。

$N$ 表示被廣播的資料物件數量，每一個資料物件在廣播通道上被一個

或多個用戶端所接收；M則是代表所有用戶端所提出的查詢種類，每個查詢最少需要廣播通道上一個資料物件；而選擇率S則是表示在一個查詢中QDS大小的百分比，例如選擇率4%即表示一個查詢需要接收4%的被廣播資料集的資料物件數；h同樣表示是插入廣播週期中熱門資料物件的數量。

前述我們利用造檔程式，造出常態型及指數型兩種實驗用資料檔，其N、M、S分別為500、200、002；隨後產生一組不重複的初始廣播序列，並且經由模擬退火法找到一組最終且較佳的序列，並且記錄其總距離之值，本初始序列將在以下各實驗中扮演重要角色，並視為開始插入熱門資料物件時的初始序列，主要是儘可能在每次實驗中建立接近的基準點。

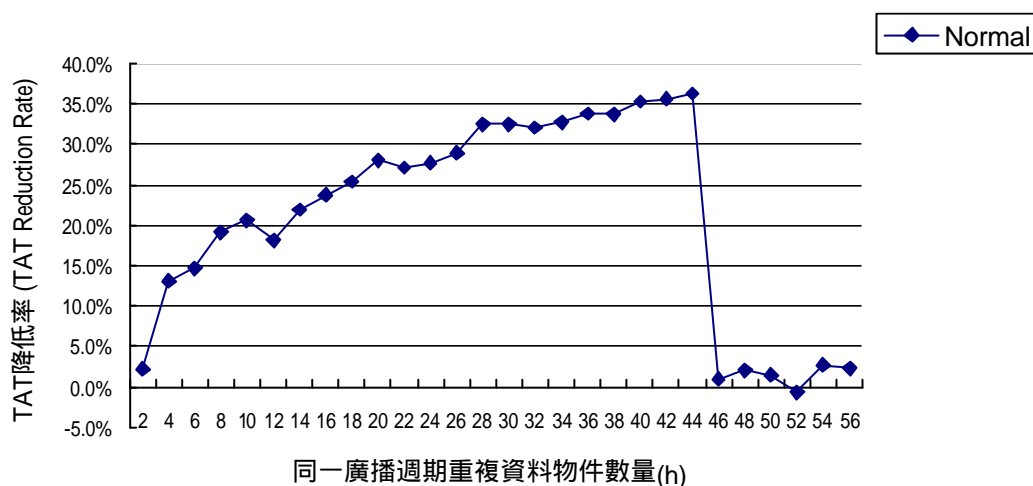
第一個實驗中，我們使用指數資料分佈的實驗資料檔，並且固定N=500、M=200及S=002，h值由2變化至56，實驗結果如圖十六，隨著開始插入熱門資料物件數量後，我們比較了開始重複後，每次的最短總距離與不重複時最短總距離的比值，意即隨著插入熱門資料數量的增加，與不重複時總距離下降與否的情形，圖中我們得到隨著每次插入的熱門資料物件數量越多，總距離逐漸下降，代表降低率越來越高，範圍約在10% 35%之間，平均24.6%，這樣的結果呈現，真實環境有些資料是相當熱門，而有些資料則是乏人問津，在同一週期中插入熱門資料，對於快速滿足用戶端需求，降低用戶端平均接收時間是有絕對幫助的。



圖十六 指數資料分佈下重複與不重複資料物件關係圖

第二個實驗中，我們使用了常態資料分佈的實驗資料檔，並且固定  $N=500$ 、 $M=200$ 及 $S=002$ ， $h$ 值由2變化至56，實驗結果如圖十七，同樣隨著開始插入熱門資料物件數量後，常態型資料檔仍然能夠呈現穩定的降低率，在此我們同樣比較了開始重複後，每次的總距離與不重複時的比值，觀察重複插入熱門資料物件時總距離下降百分比的情形，圖中我們得到隨著每次插入的熱門資料物件數量越多，總距離逐漸下降，意即TAT降低率越來越高，範圍約在2% 37%之間，這樣的結果呈現，在常態資料分佈狀態下，同一週期中插入熱門資料，對於快速滿足用戶端需求，降低用戶端接收時間是有絕對的幫助；另外再插入第46個熱門資料物件之後，降低率則急速銳減，則顯示了在插入的熱門資料物件數量中，因每個資料物件的熱門程度並不相同，可能因為資料物件的熱門程度不

佳，插入同一週期中，讓週期變長，相對於用戶端總距離反而會拉長，非但無法幫助降低用戶端接收時間，過多的熱門資料物件插入，讓用戶端接收時間變的更長，此一狀況說明了在同一週期中過多的重複資料物件，對於用戶端能否快速接收所需資料物件並無幫助，相同可能帶來負面效果。



圖十七 常態資料分佈下重複與不重複資料物件關係圖

透過圖十六與圖十七，我們了解熱門資料物件插入同一週期中，並且在兩種不同資料分佈的實驗資料檔下，觀察重複與不重複資料物件時總距離變化的情形，過程中我們使用了模擬退火演算法來幫助我們找到較佳的廣播序列，並且進行量測距離的計算。表五及表六說明了整個過程在指數及常態資料分佈下相關的實驗數據，其中原始距離代表初始序

原始距離	最短距離	插入物件數	降低率 A	降低率 B
384389	64838	0		83.1%
385445	57553	2	11.2%	85.1%
388001	56439	4	13.0%	85.5%
342501	55888	6	13.8%	83.7%
341001	53919	8	16.8%	84.2%
333824	54990	10	15.2%	83.5%
303396	54090	12	16.6%	82.2%
292663	52916	14	18.4%	81.9%
271190	51550	16	20.5%	81.0%
269860	51920	18	19.9%	80.8%
261217	52057	20	19.7%	80.1%
262518	50657	22	21.9%	80.7%
263467	49369	24	23.9%	81.3%
245520	49910	26	23.0%	79.7%
245111	49522	28	23.6%	79.8%
211417	48937	30	24.5%	76.9%
211473	47775	32	26.3%	77.4%
211446	47230	34	27.2%	77.7%
199594	46329	36	28.5%	76.8%
178467	45907	38	29.2%	74.3%
176652	45799	40	29.4%	74.1%
173383	44835	42	30.9%	74.1%
174101	43789	44	32.5%	74.8%
167609	44339	46	31.6%	73.5%
168669	43372	48	33.1%	74.3%
148554	43701	50	32.6%	70.6%
149525	41680	52	35.7%	72.1%
148423	41953	54	35.3%	71.7%
144844	41683	56	35.7%	71.2%
		平均	24.6%	78.3%

表五 指數型態資料分佈實驗數據

列或是將熱門資料以亂數方式插入至同一週期中的序列，在未經模擬退火法演算過程的距離量測值；最短距離則是序列在退火過程中量測的最短距離；降低率A代表開始插入熱門資料物件所得到的最短距離與未插入任何物件時的最短距離，中間差值的比率（如實驗一、二所述）；降低率B代表開始插入熱門資料物件所得到的原始距離與經由模擬退火過程後的最短距離，中間差值的比率。

表五中我們發現指數型態資料分佈，在開始插入熱門資料物件之後，並且未經模擬退火法擾動退火之前的原始距離是隨著熱門資料物件插入的數量越多，呈現下降趨勢，這表示熱門資料物件的插入，對於立即降低用戶端接收時間是有幫助的；至於每次的插入定量熱門資料物件並經模擬退火法後，前後所產生的差值比例，即降低率B，均維持在70% 85%之間，平均78.3%。

另表六常態型態資料分佈中，開始插入熱門資料物件之後，並且未經模擬退火法擾動退火之前的原始距離，隨著熱門資料物件插入的數量越多，並無明顯呈現下降趨勢，直插入46個熱門資料物件之後，原始距離反而開始增加，同降低率A所呈現的狀態，這說明了過多熱門資料物件的插入，可能導致的反效果現象；另外降低率B，均維持在70% 81%之間，平均77.5%，這樣的結果說明了利用模擬退火的演算方法，對於我們求得最佳的廣播序列是有正面且積極幫助的。



原始距離	最短距離	插入物件數	降低率 A	降低率 B
257337	73557	0		71.4%
242847	71949	2	2.2%	70.4%
256218	63926	4	13.1%	75.1%
254489	62786	6	14.6%	75.3%
254815	59495	8	19.1%	76.7%
251361	58399	10	20.6%	76.8%
252189	60251	12	18.1%	76.1%
255130	57434	14	21.9%	77.5%
254653	56087	16	23.8%	78.0%
253498	54911	18	25.3%	78.3%
257601	52949	20	28.0%	79.4%
254796	53619	22	27.1%	79.0%
255691	53179	24	27.7%	79.2%
256193	52304	26	28.9%	79.6%
261540	49672	28	32.5%	81.0%
260902	49703	30	32.4%	80.9%
252583	49989	32	32.0%	80.2%
252988	49467	34	32.8%	80.4%
256928	48683	36	33.8%	81.1%
257258	48775	38	33.7%	81.0%
254129	47589	40	35.3%	81.3%
254627	47360	42	35.6%	81.4%
252079	46881	44	36.3%	81.4%
280939	72844	46	1.0%	74.1%
281981	72051	48	2.0%	74.4%
283122	72498	50	1.4%	74.4%
283564	74084	52	-0.7%	73.9%
284987	71588	54	2.7%	74.9%
286060	71876	56	2.3%	74.9%
		平均	20.8%	77.5%

表六 常態型態資料分佈實驗數據

透過實驗，我們嘗試使用兩種不同資料分佈的資料庫，由初始的廣播序列中逐次插入受歡迎的資料物件，並且紀錄經由退火法之後產生的總距離，用來了解在同一廣播週期中重複資料物件，讓原先週期變長，對於用戶端所產生的變化，並且觀察它們對於真實環境中可能造成的影響，結果我們由實驗所得數據了解，對於在同一廣播週期中，插入熱門程度不同的資料物件，基本上可以明顯反映到用戶端的接收時間上，用戶端平均的接收時間下降了，但是我們並不能迷失於無限制的將資料物件重複於同一廣播週期中，它會因為每個資料物件的受歡迎程度不同，重複至同一週期時，迫使週期變長，反而不利於多數用戶端的接收，進而帶來負面的影響。另論文中我們藉由模擬退火演算法的運作原理，將廣播排程最佳化的問題類比化，更將模擬退火法各種參數設定概念移植，並且貫穿於整個找尋最佳廣播排程的過程，實驗結果呈現了模擬退火法的確可以有效幫助我們去找尋最佳的廣播排程，降低用戶端平均接收時間，

## 第五章 結論與未來研究方向

行動環境中為了實現科技結合生活的目標，滿足越來越多的消費大眾，利用無線廣播來傳遞訊息為目前最常被使用的方法。但是受限於頻寬及行動裝置電力的不足，為了快速滿足用戶端所需，縮短用戶端等待接收資料時間，目前學術上主要研究，皆著重在適當的演算法則的設計，以進行廣播資料物件排程最佳化，幫助用戶端藉由優良的廣播資料序列，快速取得所需資料。本論文在多重要求下，探討以「非單調式」模式進行廣播，對於用戶端具有正面幫助，用戶端整體的接收時間明顯下降。

科學家們利用對於大自然觀察的現象，發展出了模擬退火法，用來解決各種最佳化的問題，期望能在有限而且合理的時間範圍內，找到一個能夠為人所接受，並且能夠幫助解決問題的方法。本論文探討在非單調式的廣播週期中，運用模擬退火演算法，成功地找尋一組最佳的廣播序列，並有效的降低用戶端接收資料的平均存取時間。

我們所提出的研究範疇，是以「多重要求」與「非單調式」的廣播模式為研究環境，研究中我們得知重複的熱門資料物件達到一定數量後，對於用戶端的幫助則逐漸趨緩，甚至會引發反效果，也就是無法滿足用戶端在極短時間內接收所需資料的要求。在真實環境中，不同的應用層面或許代表著有不同數量的資料物件，資料物件的多寡以及重複數量的程度，深切影響用戶端的接收時間，未來我們殷切期

盼可以在上述環境中找到一些合宜的算式，這樣或許可以幫助系統廠商，快速找到資料物件與重複熱門資料物件數量之間的關係，而這些對業者比較具有實質意義與幫助；除此之外仍然還有很多有關無線廣播資料排程的技術需要去探勘，例如多通道無線環境、非同步無線廣播環境或者是順序性廣播頻道等研究，這些都是我們未來將持續研究的目標。

## 參考文獻

### 一、中文部份

1. 余朝權，林聰武，王政忠”資訊商品的網路行銷” 資訊管理學報，第六卷，第一期，頁25-45。
2. 吳俊樂 ”利用分段法改善線性規劃式無線廣播機制” 國立中興大學資訊科學研究所碩士班碩士論文 民91。
3. 巫沛倉、楊文宏、洪國偉 “應用遺傳演算法與模擬退火法在存貨模式之研究” 2002年科技與管理學術研討會論文集,頁 329~339.
4. 謝佳宏 “電子商務對總要素生產力之影響” 銘傳大學經濟系碩士碩士論文 民92。

### 二、西文部份

5. S. Acharya, R. Alonso, M. Franklin and S. Zdonik, “Broadcast disks: Data management for asymmetric communication environments,” in Proceedings of ACM SIGMOD Conference, pp.199-210, 1995.
6. S. Acharya, M. Franklin and S. Zdonik, ”Disseminating Updates on broadcast disk,” in Proceedings of Very Large Data Bases Conference, pp.354-365, 1996.
7. Demet Aksoy and MasonSin-FaiLeung, “Pull vs Push:A Quantitative Comparison for Data,” Communication Society Globecom IEEE, pp.1464-1468, 2004.

8. T.F. Bowen, G. Gopal, G. Herman, T. Hickey, K.C. Lee, W.H. Mansfield, J Raitz and A. Weinrib. "The datacycle architecture," *Communications of the ACM* , 35(12) , 1992.
9. A.A. Bertossi, M.C. Pinotti, S. Ramaprasad, R. Rizzi and M.V.S. Shashanka, "Optimal multi-channel data allocation with flat broadcast per channel," *Proceedings of the 18th International and Distributed Processing Symposium(IPDPS'04) IEEE*, 2004.
10. Y.D. Chung and M.H. Kim, "An index replication scheme for wireless data broadcasting," *Journal of Systems and software*, vol.51, no.3, pp.191-199, 2000.
11. Yon Dohn Chung and Myoung Ho Kim, "Effective Data Placement for Wireless Broadcast," *Division of Computer Science Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, 373-1, Kusung-dong, Yuseong-gu, Taejeon, 305-701, Korea*, 2001.
12. O.E. Demir and D. Aksoy, "Energy-Efficient Broadcast-based Event Update Dissemination," *Proceeding of Performance Computing and Communications IEEE*, pp.477-482, 2004.
13. V. Gondhalekar, "Scheduling Periodic Wireless Data Broadcast, M.S.Thesis," *The University of Texas at Austin, U.S.A.*, 1995.
14. Chih-Lin Hu and Ming-Syan Chen, "Adaptive Balanced Hybrid Data Delivery for Multi-Channel Data Broadcast," *IEEE International Conference on Communications*, vol.2, pp.960-964, 2002.
15. Chin-Hao Hsu, Guanling and Arbee L.P. Chen, "A Near Optimal

- Algorithm for Generating Broadcast Programs on Multiple Channels,” Proceedings ACM 10th Int'l Conference Information and Knowledge Management, pp.303-309, 2001.
16. Chin-Hao Hsu, Guanling and Arbee L.P. Chen, "Index and Data Allocation on Multiple Broadcast Channels Considering Data Access Frequencies," Proceedings Int'l Conference Mobile Data Management, pp.87-93, 2002.
  17. G. Herman, G. Gopal, K.C. Lee and A. Weinrib, “The datacycle architecture for very high throughput database systems,” In Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data, pp.97-103, 1987.
  18. S. Kirkpatrick et.al., “Optimization by simulated annealing”, Science 220 (1983) 671-80.
  19. T.Imielinski, S.Viswanathan and B.R. Badrinath, ”Power Efficient Filtering of Data on Air,” Proceedings of the International Conference on Extending Database Technology, pp.245-258, 1994.
  20. T. Imielinski, S. Viswanathan and B.R. Badrinath, ”Data on air Organization and access,” IEEE Transactions on Knowledge and Data Engineering, vol.9, no.3, pp.353-372,1997.
  21. T. Imielinski, S. Viswanathan and B.R. Badrinath, “Energy Efficient Indexing on Air,” in Proceedings of ACM SIGMOD Conference, pp. 25-36, 1994.
  22. Sung-Hwa Lim and J.H. Kim, “Real-time broadcast algorithm for mobile computing,” The Journal of Systems Software 69,

pp.173-181, 2004.

23. W.C. Lee and D.L. Lee, "Using Signature Techniques for Information Filtering in Wireless and Mobile Environments," *Distributed and Parallel Databases*, vol.4, no.3, pp.205-227, 1996.
24. S.C. Lo and A.L.P. Chen, "Optimal Index and Data Allocation in Multiple Broadcast Channels," *Proceedings 16th IEEE Int'l Conference Data Eng.*, pp.293-302, 2000.
25. S.C. Lo and A.L.P. Chen, "An Adaptive Access Method for Broadcast Data under an Error-Prone Mobile Environment," *IEEE Trans. Knowledge and Data Eng.*, vol.12, no.4, pp.609-620, 2000.
26. OBV Ramanaiah and Hrushikesh Mohanty, "NICD:A Novel Indexless Wireless On-Demand Data Broadcast Algorithm," *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) IEEE*, 2004.
27. C.Su, L.Tassiulas and V.j. Tsotrs, "Broadcast scheduling for information distribution," *Wireless Network*, vol.5, pp.137-147, 1998.
28. Navrati Saxena, Kalyan Basu and Sajal K. Das, "Design and Performance Analysis of a Dynamic Hybrid Scheduling Algorithm for Heterogeneous Asymmetric Environments," *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04) IEEE*, 2004.
29. K.L. Tan and J.X. Yu. "A dynamic scheduler for the infinite air-cache," *Data and Knowledge Engineering* 24, PP.97-112, 1997.



30. N. Vaidya and S. Hameed, "Scheduling Data Broadcast in Asymmetric Communication Environments," *ACM/Baltzer Wireless Networks*, vol. 5, no.3, pp.171-182, 1999.
31. K.H. Yeung and T.S. Yum, "Selective Broadcast Data Distribution Systems," *IEEE Trans. Computers*, vol.46, no.1, pp.100-104, 1997.

# 附 錄 一

/\*

系 所: 南華大學 資訊管理研究所

中文題目: 在行動環境中之可重複的資料廣播問題

英文題目: Broadcast Data Location with Replication in Mobile Environments

子 系 統: 統計熱門資料物件並依序排行

指導教授: 吳光閔 博士

研 究 生: 王子修

使用語言: Microsoft Visual C++ 6.0

範例檔案: statistice.cpp

資料分佈: normal && exponential

製作日期: 2005/02/05

\*/

```
#include <iostream.h>
```

```
#include <fstream.h>
```

```
#include <stdlib.h>
```

```
#include <iomanip.h>
```

```
#include <algorithm>
```

```
#include <time.h>
```

```
#include <math.h>
```

```
#define Dos 500 //data objects 數量
```

```
#define QS 200 //Queries 數量
```

```
#define Sel 10 //select 數量
```

```
unsigned int A_ary[DOs][2]; //統計陣列
```

```
void main()
```

```
{
```

```
    char fn[20]="ed500q200s002.txt";
```

```

int    i, j, l, num, val, temp[2];
// ----- 讀入資料 -----
for (i=0; i<DOs; i++) { A_ary[i][0]=i; A_ary[i][1]=0; }
fstream file;
file.open(fn, ios::in);
if(!file) { cerr << "Can't open file!\n"; exit(1); }
for (i=0; i<QS; i++)
{   for (j=0; j<=Sel; j++)
    {   if (j==0) file>>num;
        else
            {   file>>val;    A_ary[val][1]+=num; }
    }
}
file.close();
// ----- 利用插入排序法將熱門資料由熱門往冷門排列 -----
for (i=1; i<DOs; i++)
{
    for (j=0; j<2; j++) temp[j]=A_ary[i][j];
    l=i;
    while (l>0 && A_ary[l-1][1]<temp[1])
    {
        for (j=0; j<2; j++) { A_ary[l][j]=A_ary[l-1][j]; }
        --l;
    }
    for (j=0; j<2; j++) A_ary[l][j]=temp[j];
}
// ----- 統計完成結果寫入 popular_data.txt -----

```

```
ofstream fop;  
fop.open("popular_data.txt", ios::out|ios::app);  
if(!fop) { cerr << "Can't open file!\n"; exit(1); }  
for (i=0; i<DOs; i++) fop<<A_ary[i][0]<<" ";  
fop.close();  
}
```



## 附 錄 二

/\*

系 所: 南華大學 資訊管理研究所

中文題目: 在行動環境中之可重複的資料廣播問題

英文題目: Broadcast Data Location with Replication in Mobile Environments

子 系 統: 不重複資料物件條件下距離計算

指導教授: 吳光閔 博士

研 究 生: 王子修

使用語言: Microsoft Visual C++ 6.0

範例檔案: bdlrme\_single.cpp

資料分佈: normal && exponential

製作日期: 2005/02/05

\*/

```
#include <iostream.h>
```

```
#include <fstream.h>
```

```
#include <stdlib.h>
```

```
#include <iomanip.h>
```

```
#include <algorithm>
```

```
#include <time.h>
```

```
#include <math.h>
```

```
#define Dos 500 //data objects 數量
```

```
#define QS 200 //Queries 數量
```

```
#define Sel 10 //select 數量
```

```
#define K 12000 //每一溫度位階擾動 K 次
```

```
unsigned int QUERY_ary[QS][Sel+1]; //所有用戶端所需查詢資料
```

```
unsigned int A_ary[DOs] B_ary[DOs]; //前後所用資料序列 A 及 B
```

```
unsigned int cost_cau(unsigned int *, unsigned int (*Q)[Sel+1]); //量測函數原型
```

```

void main()
{
    char  fn[20]="ed500q200s002.txt";           //實驗資料庫
    int   i, j, l, d1, s_d1, d2, r1, r2, Delta_l, gmin=999999;
    double e=2.718;
    double t=1000.0, q, r3;
    // ----- 開啟檔案並讀入 QUERY_ary -----
    ifstream file;
    file.open(fn, ios::in);
    if(!file) { cerr << "Can't open file!\n"; exit(1); }
    for (i=0; i<QS; i++)
    {
        for (j=0; j<=Sel; j++)
            (j==0)? file>>QUERY_ary[i][Sel]:file>>QUERY_ary[i][j-1];
    }
    file.close();
    // ----- 產生初始序列並寫入 inzi_sch.txt -----
    ofstream dop;
    dop.open("inzi_sch.txt", ios::out|ios::app);
    if(!dop) { cerr << "Can't open file!\n"; exit(1); }
    srand((unsigned)time(NULL));               //產生最初的一組序列
    for (i=0; i<DOs; i++) { A_ary[i]=0; B_ary[i]=0;} //擷取時間為亂數因子
    for (i=0; i<DOs; i++)                       //採用線性探測 linear probing 修正碰撞
    {
        l=rand()%DOs; while (A_ary[l]!=0) { l++; l%=DOs; }
        A_ary[l]=i; dop<<A_ary[i]<<" "; B_ary[l]=i;
    }
}

```

```

d1=cost_cau(A_ary, QUERY_ary); s_d1=d1;
dop.close();
// ----- simulated annealing operator 模擬退火法 -----
while (t>=1)          //溫度控制
{
    for (i=0; i<K; i++)      //擾動次數控制
    {
        if (rand()%2==0)
        {
            r1=rand()%DOs; r2=rand()%DOs;          //擾動方法一
            while (r1==r2) r2=rand()%DOs;
            l=B_ary[r1]; B_ary[r1]=B_ary[r2]; B_ary[r2]=l;
        }
        else
        {
            r1=rand()%DOs; r2=rand()%DOs;          //擾動方法二
            while (r1==r2) r2=rand()%DOs;
            if (r1<r2)
            {
                l=B_ary[r1];          //插入後段位置
                for (j=0; j<(r2-r1); j++) B_ary[r1+j]=B_ary[r1+j+1];
                B_ary[r2]=l;
            }
            else
            {
                l=B_ary[r1];          //插入前段位置
                for (j=0; j<(r1-r2); j++) B_ary[r1-j]=B_ary[r1-j-1];
            }
        }
    }
}

```

```

        B_ary[r2]=l;
    }
}
d2=cost_cau(B_ary, QUERY_ary);    //計算擾動後的距離計算
if (d2<gmin) gmin=d2;
Delta_l=d1-d2;                    //檢視擾動後的結果
if (Delta_l>=0)                    //擾動後優於擾動前無條件接受
{ for (j=0; j<DOs; j++) A_ary[j]=B_ary[j]; d1=d2; }
else
{
    q=pow(e,((double)Delta_l/t));    //條件計算
    r3=(double)(rand()%10000)/10000;
    if (q>=r3)
    { for (j=0; j<DOs; j++) A_ary[j]=B_ary[j]; d1=d2; } //接受
    else
    { for (j=0; j<DOs; j++) B_ary[j]=A_ary[j]; } //不接受
}
}
t*=0.95;                          //溫度下降公式
}
// ----- 紀錄相關結果 -----
ofstream fop;
fop.open("result_file.txt", ios::out|ios::app);
if(!fop) { cerr << "Can't open file!\n"; exit(1); }
fop<<fn<<" "<<"原始距離="<<s_d1<<" "<<"最短距離="<<d1<<" "<<"最短距離="<<gmin<<"\n";
fop.close();

```



```

}
// ----- 單調式距離量測函數 -----
unsigned int cost_cau(unsigned int *b_ptr, unsigned int (*q_ptr)[Sel+1])
{
    unsigned int r, x, w, y, z; unsigned int distance=0;
    unsigned int c_ary[DOs], dis_ary[DOs]; bool result;
    for (x=0; x<QS; x++)
    {
        for (y=0; y<DOs; y++) { c_ary[y]=0; dis_ary[y]=0; }
        for (y=0; y<Sel; y++)
        {
            z=0; result=true;
            while (result)
            {
                if ( (*(q_ptr+x)+y)==*(b_ptr+z) ) { c_ary[z]=1; result=false; }
                else z++;
            }
        }
        z=0; w=DOs-1;
        while (!(c_ary[z] && c_ary[w]))
        { if (c_ary[z]==0) z++; else if (c_ary[w]==0) w--; }
        for (r=0; r<(DOs-(w-z+1)); r++) dis_ary[r]=1;
        for (r=z; r<w; r++) { if (c_ary[r]==1) z=0; else {dis_ary[z]=1; z++;} }
        r=0;
        while (dis_ary[r]==1) r++;
        distance+=(DOs-r)*(*(q_ptr+x)+Sel);
    }
}

```

```
    return distance;  
}
```

## 附 錄 三

/\*

系 所: 南華大學 資訊管理研究所

中文題目: 在行動環境中之可重複的資料廣播問題

英文題目: Broadcast Data Location with Replication in Mobile Environments

子 系 統: 重複熱門資料物件條件下距離計算

指導教授: 吳光閔 博士

研 究 生: 王子修

使用語言: Microsoft Visual C++ 6.0

範例檔案: bdl sme\_Replication.cpp

資料分佈: normal && exponential

製作日期: 2005/02/12

\*/

```
#include <iostream.h>
```

```
#include <fstream.h>
```

```
#include <stdlib.h>
```

```
#include <iomanip.h>
```

```
#include <algorithm>
```

```
#include <time.h>
```

```
#include <math.h>
```

```
#define PN 56 //熱門資料物件數量(2-56)
```

```
#define Dos 500 //data objects 數量
```

```
#define QS 200 //Queries 數量
```

```
#define Se 10 //select 數量
```

```
#define K 12000 //每一溫度位階擾動 K 次
```

```
unsigned int QUERY_ary[QS][Sel+1]; //所有用戶端所需查詢資料
```

```
unsigned int A_ary[DOs+PN], B_ary[DOs+PN]; //前後所用資料序列 A 及 B
```

```
unsigned int cost_cau(unsigned int *, unsigned int (*Q)[Sel+1]); //量測函數原型
```

```
void main()
```

```
{
```

```
    char fn[20]="ed500q200s002.txt";           //讀入查詢用戶端資料檔
```

```
    char is[15]="inzi_sch.txt";                //讀入初始廣播排程
```

```
    char pu[20]="popular_data.txt";           //讀入熱門資料
```

```
    int i, j, l, d1, s_d1, d2, r1, r2, Delta_l, gmin=999999;
```

```
    double e=2.718;
```

```
    double t=1000.0, q, r3;
```

```
    // ----- 開啟檔案並讀入 QUERY_ary -----
```

```
    fstream file_a;
```

```
    file_a.open(fn, ios::in);
```

```
    if(!file_a) { cerr << "Can't open file_a!\n"; exit(1); }
```

```
    for (i=0; i<QS; i++)
```

```
    {
```

```
        for (j=0; j<=Sel; j++)
```

```
            (j==0)? file_a>>QUERY_ary[i][Sel]:file_a>>QUERY_ary[i][j-1];
```

```
    }
```

```
    file_a.close();
```

```
    // ----- inserting popular data objects -----
```

```
    for (i=0; i<DOs+PN; i++) { A_ary[i]=0; B_ary[i]=0; }
```

```
    fstream file_b;           //開啟讀入起始廣播序列
```

```
    file_b.open(is, ios::in);
```

```
    if(!file_b) { cerr << "Can't open file_b!\n"; exit(1); }
```

```
    for (i=0; i<DOs; i++) { file_b>>A_ary[i]; B_ary[i]=A_ary[i]; }
```

```
    file_b.close();
```

```
    srand((unsigned)time(NULL));           //亂數
```

```

fstream file_c;           //開啟讀入熱門資料序列
file_c.open(pu, ios::in);
if(!file_c) { cerr << "Can't open file_b!\n"; exit(1); }
for (i=1; i<=PN; i++)     //將熱門資料以亂數方式插入廣播序列
{
    d1=DOs+i; file_c>>l; r1=rand()%(DOs+i-1);
    for (j=d1-2; j>=r1; j--) { A_ary[j+1]=A_ary[j]; B_ary[j+1]=B_ary[j]; }
    A_ary[r1]=l; B_ary[r1]=l;
}
file_c.close();
d1=cost_cau(A_ary, QUERY_ary); s_d1=d1;           //距離計算
// ----- simulated annealing operator 模擬退火法 -----
while (t>=1)           //溫度控制
{
    for (i=0; i<K; i++)       //擾動次數控制
    {
        if (rand()%2==0)
        {
            r1=rand()%(DOs+PN); r2=rand()%(DOs+PN);   //擾動方法一
            while (r1==r2) r2=rand()%(DOs+PN);
            l=B_ary[r1]; B_ary[r1]=B_ary[r2]; B_ary[r2]=l; //計算 COST 值
        }
        else
        {
            r1=rand()%(DOs+PN); r2=rand()%(DOs+PN);   //擾動方法一
            while (r1==r2) r2=rand()%(DOs+PN);
            if (r1<r2)

```

```

{
    l=B_ary[r1];          //插入後段位置
    for (j=0; j<(r2-r1); j++) B_ary[r1+j]=B_ary[r1+j+1];
    B_ary[r2]=l;
}
else
{
    l=B_ary[r1];          //插入前段位置
    for (j=0; j<(r1-r2); j++) B_ary[r1-j]=B_ary[r1-j-1];
    B_ary[r2]=l;
}
}
d2=cost_cau(B_ary, QUERY_ary);
if (d2<gmin) gmin=d2;
Delta_l=d1-d2;          //檢視擾動後的結果
if (Delta_l>=0)          //擾動後優於擾動前無條件接受
{ for (j=0; j<DOs+PN; j++) A_ary[j]=B_ary[j]; d1=d2; }
else
{
    q=pow(e,((double)Delta_l/t));          //條件計算
    r3=(double)(rand()%10000)/10000;
    if (q>=r3)
    { for (j=0; j<DOs+PN; j++) A_ary[j]=B_ary[j]; d1=d2; } //接受
    else
    { for (j=0; j<DOs+PN; j++) B_ary[j]=A_ary[j]; } //不接受
}
}
}

```

```

        t*=0.95;          //溫度下降公式
    }
// ----- 紀錄相關結果 -----
ofstream fop;
fop.open("result_file.txt", ios::out|ios::app);
if(!fop) { cerr << "Can't open file!\n"; exit(1); }
fop<<"PN="<<PN<<" "<<fn<<" "<<" 原始距離 ="<<s_d1<<" "<<" 最短距離
="<<d1<<" "<<" 最短距離="<<gmin<<"\n";
fop.close();
}
// ----- 非單調式距離量測函數 -----
unsigned int cost_cau(unsigned int *b_ptr, unsigned int (*q_ptr)[Sel+1])
{
    int x, y, z, flag, lc, r, len, len_min, old;          //變數定義
    int distance=0;
    int b[Sel], ad[DOs];          //ad 紀錄位置 query 的兩倍
    for (x=0; x<QS; x++)
    {
        for (y=0; y<Sel; y++) b[y]=0;
        for (y=0; y<DOs; y++) ad[y]=999;          //清空及設定 b ad 陣列
        lc=0;          //幫助記憶位置
        for (y=0; y<DOs+PN; y++)          //統計次數並且紀錄位置
        {
            flag=0; z=0;
            while (flag==0 && z<Sel)
            {
                if (*(b_ptr+y)==*(q_ptr+x+z)) { b[z]+=1; flag=1; }
            }
        }
    }
}

```

```

        else z++;
    }
    if (flag==1) { ad[lc]=y; lc++; }
}
y=0;          //找出 l 及 r 值(左右兩邊指標)
while (ad[y]<999)
{
    if (y==0) { lc=ad[y]; y++; }
    else y++;
}
if (*(b_ptr+ad[y-1])==*(b_ptr+ad[y-2])) { r=ad[y-2]; ad[y-1]=999; }
else r=ad[y-1];
y=0; len_min=999999;          //計算最小的長度
while (ad[y]<999)
{
    lc=ad[y]; len=r-lc+1; if (len<len_min) len_min=len;
    y++; old=*(b_ptr+lc); z=0; flag=0;
    while (flag==0)
    {
        if (*(q_ptr+x)+z==old)
        {
            b[z]-=1;
            if (b[z]==0)
            {
                r++;
                while (*(b_ptr+r%(DOs+PN))!=old) r++;
                flag=1;
            }
        }
    }
}

```



```
        b[z]+=1;
    }
    else { flag=1; b[z]+=1; }
    }
    else z++;
}
}
distance+=len_min*(*(q_ptr+x)+Sel));
}
return distance;
}
```