國 立 成 功 大 學

工 業 與 資 訊 管 理 研 究 所

碩 士 論 文

分別以連鎖不平衡及拉氏鬆弛法選取代表性單核苷酸多型性之研究

TagSNP Selection Problems based on Linkage Disequilibrium and Lagrangian Relaxation

指導教授：王逸琳 老師

研 究 生：馬家宜

中 華 民 國 九 十 七 年 八 月

國立成功大學
碩士論文

分別以連鎖不平衡及拉氏鬆弛法
選取代表性單核苷酸多型性之研究

TagSNP Selection Problems based on Linkage Disequilibrium and Lagrangian Relaxation

研究生：馬家宜

本論文業經審查及口試合格特此證明

論文考試委員：王逸琳

蔡孟志

陳一薰

高耀廷

李宇欣

指導教授：王逸琳

系(所)主管：李俊璋

中 華 民 國 97 年 5 月 30 日

摘要

在DNA序列可能發生的眾多差異性當中，單核苷酸多型性（Single Nucleotide Polymorphism，SNP）是最常發生的一種遺傳變異，由多個SNP鹼基所組成之序列稱為基因組單體形（Haplotype），此序列的改變對疾病的發生及人類特徵的顯現有重大關聯，可被應用於辨識不同疾病及其他相關之醫學研究上。由於目前已發現的SNP資料量龐大，為了節省SNP資料庫所需的高成本花費，許多研究建議以被稱為tagSNP的SNP序列資料部份集合來代表原本全部的SNP序列。

tagSNP可依其應用目的而有不同的定義，本研究首先針對文獻中最常被使用之tagSNP定義，將其選取問題(Selection Problem) 轉換成一個具有多重最佳解的0,1二元整數規劃問題，以選出可辨識出所有的Haplotype序列樣式之最小SNP部分集合。由於過去研究多著重於改善求解方法之效率，並未評估所求得之最佳解與其它最佳解間之資訊差異，因此本研究提出一個以圖型理論為基礎的啟發式演算法，先求解出所有的最佳解，再採用連鎖不平衡(Linkage Disequilibrium，LD)觀念以計算已被選取之tagSNP與尚未被選出之其它SNP間的相互關連性，作為該最佳解所包含Haplotype資訊量多寡的評估指標，並選取其中最多資訊者為最終最佳解。此外，我們亦提出一個可同時考慮極小化tagSNP個數與極大化LD值之和的雙目標數學規劃模式以求解類似問題。

在求解大規模的tagSNP選取問題上，本研究提出一個以拉氏鬆弛法為基礎的啟發式演算法LRH，採用次梯度(subgradient)法更新拉氏乘數(Lagrangian multiplier)，使之逐漸逼近最佳解。我們亦在求解過程中加入貪婪演算法的觀念，藉由固定部份SNP欄位以逐漸縮減問題規模，改善求解速度及求解品質。此外，我們亦提出一個結合LRH與最佳化軟體CPLEX兩者優點的二階段求解方法；數值測試結果顯示該二階段解法的確可以在更短的時間內選取出品質更佳之tagSNP解。最後，本研究提出一個整數規劃模式以在具有容量限制的生物晶片上選取較可靠的tagSNP解，並呈現容量限制下限與辨識之可靠性間的關係圖以供後續研究參考。

關鍵字：標記型單核苷酸多型性、連鎖不平衡、基因組單體型、演算法、拉氏鬆弛法

Abstract

Among all possible DNA sequence variations, Single Nucleotide Polymorphism（SNP）is the most common genetic variation. Sequence of closely linked SNPs composes the haplotype. Changes in the sequence can have significant influence on disease occurrences and the phenotype of human traits. The changes can be applied for identifying diseases and other medical research. At present, there are voluminous data of discovered SNPs. To make SNP database more cost-effective, many researchers propose tagging SNPs, a minimal subset of SNPs called tagSNP, to capture the full information of the original SNP sequence（haplotype）.

Various SNP tagging methods have been proposed in the literature, based on different purposes of applications. This paper focuses on the most commonly cited definition of tagSNP and proposes methods to select them. In particular, we first seek the smallest SNP subset to identify all Haplotype patterns. The tagSNP Selection Problem can be modeled as a 0-1 binary integer programming problem with multiple optimal solutions. Previously, scholars focused on improving the efficiency of their solution methods, and ignored the differences among multiple optimal solutions. To this end, we proposes a Heuristic algorithm based on graph theory that first solves for all the multiple optimal solutions and then recommends a more informative set of optimal solution based on the concept of Linkage Disequilibrium（LD）. Our method calculates the relevancy between a selected tagSNP and the other SNPs, which later serves as an indicator for assessing the amount of information it carries. Among all the multiple optimal solutions, we recommend the one with the largest sum of LD values. In addition, we also propose a bi-objective integer programming model which tries to minimize the number of selected tagSNPs while maximize the sum of their LD values.

To deal with large-scale tagSNP selection problems, we propose a heuristic called

LRH, based on the theory of Lagrangian Relaxation. In particular, a modified subgradient method is proposed to update the Lagrangian multiplier which in turn approaches to the optimal solution step by step. In LRH, we incorporate the concept of Greedy Algorithm which selects some good SNP column, gradually reduces the problem size, and thus greatly improves the efficiency and quality. The computational results show that LRH has good efficiency and effectiveness, since it can quickly converges to a better solution. Moreover, we also give a two-stage solution method（MIX） which first uses LRH to select good candidate SNP columns and then solve a reduced tagSNP selection problem of much smaller size based on the selected candidate SNPs. The computational results show that the proposed two-stage approach converges to a better solution in a much shorter time.

Finally, in a suggested future research topic, we consider the case where the selected tagSNPs is to be included in a biochip, while the capacity for the biochip is sufficiently large. In this case, we no longer have to select the SNPs of minimum size. Instead, we focus on selecting those tagSNPs that can be used to differentiate haplotypes as many as possible so that the selected SNPs are more robust or reliable.

Kayowrd：tagSNP 、Linkage Disequilibrium、Haplotype、Algorithm、Graph、Lagrangian Relaxation

誌謝

本論文得以順利完成，承蒙指導教授 王逸琳老師對論文研究及專業領域之知識給予細心的指導，老師不厭其煩地指正學生的缺失和不足，使學生的論文能臻至完備，更是讓學生領略到作學問所需的嚴謹與態度，非常值得學生去學習與效法。除此之外，在日常生活與待人處事上，感謝老師所給予的諄諄教誨，教導我們所需的正確觀念與態度，再再使學生受益匪淺。

在口試期間，感謝 黃耀廷教授、 李宇欣教授、 洪一薰教授以及 蔡青志教授對學生的論文惠賜寶貴的建議，點出論文的缺失、不足與改進的方向，讓學生的論文更具內涵與價值。

在學校修課期間，感謝 61205 最佳化演算法實驗室伙伴群達、姿君、橙坤、姿儀、正楠、建傑、俊賢與志偉的陪伴和鼓勵，讓兩年來的研究生生活過得非常充實快樂。另外，感謝同學修政、詠新、晉毅與永祥在課餘時間帶給我許多歡樂，讓我在陷入低潮時可以轉換心情，快速充電繼續向前。此外，特別感謝姿儀在研究陷入困境或是心情煩悶時聽我申訴抱怨，給予安撫與建議，讓我更有繼續努力的動力與勇氣。

最後，要感謝我的爸爸、媽媽與哥哥，這些日子的忙碌與勞心讓我的心情較起伏不定，感謝您們這段期間的包容，時時關懷著我陪我一起努力，讓我求學過程能無後顧之憂，希望我努力的成果，能夠讓您們為我驕傲。最後謹以這篇論文答謝摯愛的家人與朋友!

目錄

表目錄

圖目錄

第一章 緒論

1.1 研究背景

生物資訊學是近年來一門新興的研究領域，它結合了生物、資訊、應用數學等多種研究領域，頗受國內外學術界重視。此領域能在短期之內竄起而成為當代的顯學，其最大的推動力來自於人類基因組解讀計畫（Human Genome Project）的完成。在基因組分析的相關研究剛開始發展的時候，如何儲存 DNA、RNA 及蛋白質序列等各式各樣的生物資料庫為當時生物資訊學的發展重點；其後，研究焦點即移轉至如何由資料庫中尋找出有用的資訊並藉由分析及解釋核苷酸序列、蛋白質序列等資訊，以找出影響疾病發生的不正常變異所在。近年來許多學者致力於探究基因序列中單核苷酸多型性（Single Nucleotide Polymorphism，SNP）對人類疾病的發生與特徵性向的表現之影響（Nowotnyand et al.，2001；Shastry，2002）。相關文獻顯示 SNP 是影響疾病與特徵發生的主因，因此本研究將針對 SNP 作探討，為方便闡述本論文及研究主題，以下我們將先針對一些相關的生物資訊等專有名詞加以介紹。

1.1.1 DNA、變異（variation）、突變（Mutation）

去氧核醣核酸（Deoxyribo Nucleic Acid，DNA）是一種呈現雙股螺旋結構的遺傳因子，存在於各種生物細胞的染色體上，其主要功能在於傳達影響生物體的發育及生命機能之運作的指令。這些帶有遺傳訊息的 DNA 片段稱為基因（Gene），而生物細胞中所有的遺傳訊息通稱為基因組（Genotype）。DNA 的基本單位由四種不同的核苷酸鹼基（locus）組成，分別為腺嘌呤（adenine，A）、胸腺嘧啶（thymine，T）、胞嘧啶（cytosine，C）、鳥糞嘌呤（guanine，G）。這些鹼基呈現互補性配對，且兩種不同的鹼基對將以不同的氫鍵數目結合。在一般情況下 A＝T 由兩條氫鍵相連，而 C≡G 由

三條氫鍵相連。由於 A 與 T 間的氫鍵數較少，較易受外來因素打斷而發生變異，例如：以 T 取代 C 的變異約佔所有生物體變異發生的三分之二。

人類的 DNA 序列約有 99%是一致的，但個體間特徵的差異仍很大，這些差異是由於在生物群體的演化過程中，部分生物體細胞之細胞核內的部份基因發生變異（genetic variation）所致，譬如 DNA 的構造或排列方式發生改變等。倘若這些變異使生物性狀改變，則稱之為突變（mutation）。一般而言，突變通常對生物體是有害的，但若突變的發生使得生物個體更能適應或反應環境的變遷，則會透過「適者生存、不適者淘汰」的篩選過程中被保留下來，使得同物種間有了新的表現個體，經由群體的交配、繁衍造成群體內各個個體之間有所差異，這些差異稱之為生物變異（variation）。簡單來說，突變僅是生物體產生變異的一小部份，在同物種中，發生突變的機率很低，但產生變異的機率卻很高。

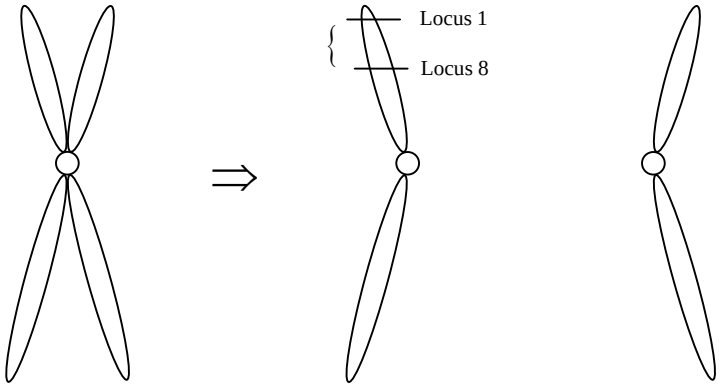
1.1.2 SNP、基因組單體型(Haplotype)

在正常細胞上，染色體（chromosome）以成對（一條來自於父親，另一條來自於母親）的方式存在，因此稱為雙套染色體。染色體由 DNA 序列構成，不同個體的 DNA 序列各有不同。在 DNA 序列上單一核苷酸鹼基對（base pair）發生的變異（亦即鹼基上 A、T、C、G 的改變），是一種常見的遺傳變異，稱之為單核苷酸多型性（Single Nucleotide Polymorphism，SNP），而此鹼基對於染色體上之對應位置稱為基因座（gene locus）。SNP 的發生率頻繁，至少佔總 DNA 序列的 0.1%，目前已發現的 SNP 總共約有 400 萬個左右，由此可知，處理如此龐大的 SNP 資料量將消耗不少成本。

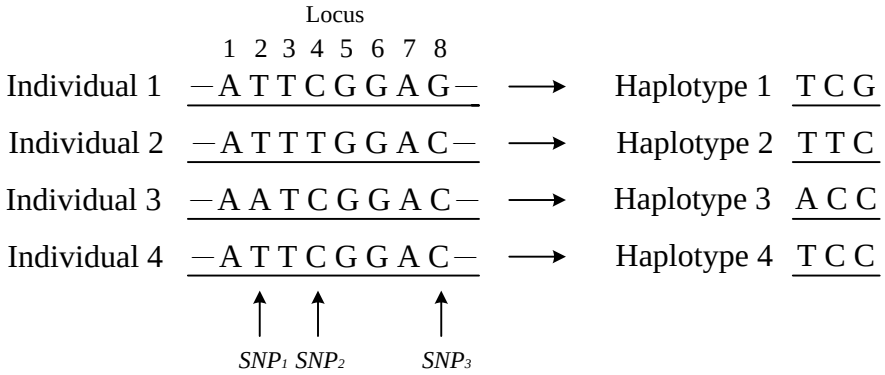
在同一族群之中不同個體間存在著相當多的變異，其中 SNP 所引起的基因變異約佔人類遺傳基因所有變異中的 90%。這些變異影響個性性狀的差異，造成生物的多樣性，且物種越接近的，其差異程度越小。舉例來說，不同人種間之 SNP 的數量及

分布較不一致，因此可能造成某些人種或群體較易產生某類疾病；譬如地中海貧血較易發生在地中海、中東、印度洋及南中國海沿岸一帶之族群，而鐮刀型貧血則普遍存在非洲黑人部落、印度土著等。

單套染色體上相鄰的 SNP 鹼基所組成之序列稱之為基因組單體型（Haplotype），以將圖 1.1（a）為例，其雙套染色體可拆成兩條單套染色體，假若擷取四個不同個體其在 chromosome 1a 中的 locus 1 到 locus 8，我們可得到圖 1.1（b）中的四條 DNA 序列。從圖中可觀察出 locus 2,4,8 在不同個體上之鹼基各有不同，因此該 DNA 序列具有 3 個 SNP，而這些 SNP 可形成一組 Haplotype 序列。舉例來說，圖 1.1（b）在右方所顯示的四個 Haplotype 序列分別為{TCG}、{TTC}、{ACC}及{TCG}。



（a）雙套染色體與單套染色體



（b）SNP與Haplotype

圖 1.1：SNP 與 Haplotype 之說明

1.2 研究動機與目的

不同個體之 DNA 序列會有所差異，此差異稱之為基因體的多型性（Genetic Polymorphism）。多型性的種類繁多，如限制脢片段長度多型性（Restriction Fragment Length Polymorphism，RFLP）、人類淋巴球抗原（Human Leukocyte Antigen，HLA）等等；其中，SNP 是 DNA 序列上最常見的一種遺傳變異，由多個 SNP 鹼基所組成的 Haplotype 可應用於病理研究上，因為 Haplotype 的不同可能造成疾病的發生、不同個體對藥物的反應不一、器官移植時的排斥作用，以及免疫系統對於不同病毒的抵抗力不同等等現象；因此，分析 Haplotype 序列的差異，將在防治疾病以及偵測潛在疾病的發生等醫療防治方面有不少貢獻。

目前在生物科技產業上十分重視生物晶片的發展，該晶片結合了半導體技術，使其每秒可進行上萬次的生化試驗，加上其體積輕巧、使用試劑量少、可大量平行處理及可拋棄等多項優點，使其在生物技術研究上的應用十分廣泛，如：開發新藥或篩選藥物、檢驗基因、醫療試驗、檢測病毒或疾病以及個人化醫療等。

由於目前已發現的 SNP 資料龐大，但生物晶片的容量有限，不可能將所有的 SNP 資料存入晶片中做應用，再加上相關研究亦指出我們其實不需要儲存如此大規模的 SNP 資料；相反地，僅由挑選出這些資料的一小部分，即可代表原始資料所欲呈現的大部分資訊。此乃由於鹼基的突變會使受驗者鹼基上的表現異於正常個體，而藉由篩選 SNP 的過程來發現這些特殊的鹼基即可找出真正影響疾病的鹼基。經由不同方式所選取出的代表性 SNP 稱之為 tagSNP，其功能乃以部分的 SNP 序列來代表原本完整的 SNP 序列，可以有效地減少資料儲存空間，並以更短的時間進行研究分析，如此便能使其在生物晶片上做應用。是故，選取 tagSNP 之問題即成為一個重要的研究議題。

1.3 研究問題

在生物資訊中，tagSNP 可依其用途不同而有不同定義，本研究針對最常被使用之定義，將 tagSNP 定義為可辨識所有不同 Haplotype 樣式的 SNP；此 tagSNP 選取問題旨在由給定的 SNP 資料中選取最少個數的 SNP 部分集合，以辨識出全部的 Haplotype 樣式。如此一來，未來若欲辨識某一個體時，可直接用所選取之 tagSNP 來快速比對，而不用再對原來所有的 Haplotype 樣式一一比對。

舉例來說，如圖 1.2 所示，假設資料庫中有 4 筆包含 10 個 SNP 之 Haplotype 資料（h_1,h_2,h_3,h_4），如欲檢查某個體的 Haplotype 資料 h_i 是否存在於該資料庫中，或是搜尋 h_i 在資料庫中的對應位置，以往必須將 h_1,h_2,h_3,h_4 一一比對方可得知（以此例而言必須比對 $4\times10=40$ 次）。然而，若能事先選取出如圖 1.2 中的 SNP_4 與 SNP_9 之 tagSNP 的話，由於其可完整辨識出 h_1,h_2,h_3,h_4，因此我們即可將 h_i 的對應位置 SNP_4 與 SNP_9 與此 tagSNP 快速比對（以此例而言，僅需比對 $2\times4=8$ 次）而得知該個體 h_i 其實已存在於資料庫中（即 h_3）。反之，若把 SNP_1 與 SNP_6 當成 tagSNP，則我們將無法得知 h_i 究竟應為是 h_2 或是 h_3。

因此，若以 tagSNP 來代表原本完整的 Haplotype 序列，將可大幅度地減少資料庫的儲存空間花費，並提昇序列比對的效率。

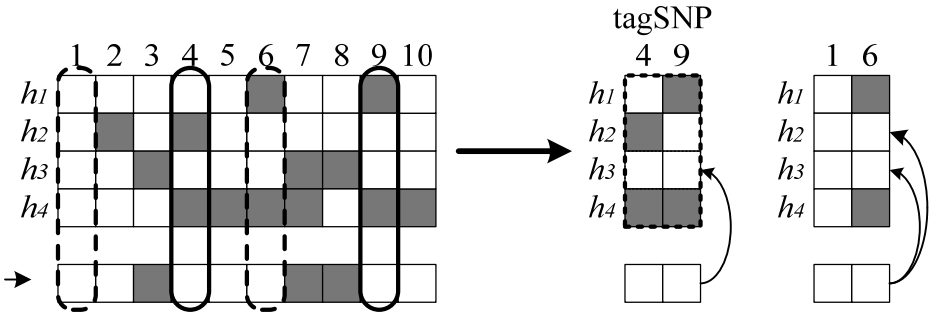


圖 1.2：以 tagSNP 辨識 Haplotype 樣式

tagSNP 選取問題通常存在多重最佳解，譬如圖 1.2 之範例除可將 SNP_4 與 SNP_9 當成 tagSNP 外，SNP_4 與 SNP_6 或是 SNP_6 與 SNP_7 亦為該問題之多重最佳解。然而，這些不同的選取結果可能代表不同的生物意義，因此本研究希望能從這些多重最佳解中選出其與未被選出之 SNP 間的彼此關連程度最高者；亦即所選取的 tagSNP 除可滿足其個數為最少之條件外，亦需涵蓋較多未選取之 SNP 的資訊（也就是選出與未被選取之 SNP 間有較高的關連程度值之 SNP）。由於此種選取結果將包含較多 SNP 資訊，因此可用於推論出整體的 Haplotype 序列，或被用來做基因的檢測與分析。在此議題上，我們首先針對規模較小的 tagSNP 選取問題提出一個啟發式演算法以選取最少個數的 tagSNP，並在所有可能的多重最佳解中選取其與未被選取的 SNP 間關連程度最高者(代表包含最多 Haplotype 序列資訊)。此外，我們亦提出一個多目標最佳化數學模式，說明如何將 tagSNP 個數極小化與該 tagSNP 所包含之序列資訊極大化同時列入考慮。

針對求解僅考慮極小化所選取之 tagSNP 個數以辨識全部 Haplotype 樣式的大規模 tagSNP 選取問題而言，本研究提出一個以拉氏鬆弛（Lagrangian Relaxation）法為基礎的啟發式演算法，該演算法可利用貪婪法則與問題特性來加速收斂所求得之解；另外，我們由數值測試的結果觀察該演算法的收斂特性，提出一個結合拉氏鬆弛法與最佳化軟體 CPLEX 優點的兩階段求解方法，更進一步加速求解效率及改善求解品質。

最後，被選取出的 tagSNP 可儲存於生物晶片中，而生物晶片的容量應在選取 tagSNP 之前已知，且其大小通常足以塞入可辨識所有 Haplotype pairs 的 tagSNPs。在該容量必定足夠可行的假設之下，如何選取最少個 tagSNP 反倒不是主要問題，此時應該考量的反而是究竟該如何利用那些多餘的容量以使所選取之 tagSNP 有更大的功能。在這方面，本論文將可能發生的生物試驗失敗所導致的資訊遺漏情況列入考慮，提出一個整數規劃模式極大化每一個 Haplotype pair 至少可被辨識的次數，以使所選取出的 tagSNP 即使在部分資訊遺漏情況下仍可達到辨識 Haplotype pair 的目的，亦即

所選取之 tagSNP 具有較高之可靠度（reliability）或強健性（robustness）。

1.4 論文架構

本論文第二章將統整過去學者針對 tagSNP 選取問題至目前為止的研究成果，包括相關研究議題之重要概念與定義，並回顧其數學模式與相關解法；第三章之研究目的在於求解 tagSNP 選取問題之多重最佳解，提出一個以圖形理論為基礎的演算法求解，並加入新的評選準則以選取出多重最佳解中能夠傳達更多訊息者，將此演算過程命名為 Multi-TagSNP 演算法；由於考慮兩個不同選取準則，因此於第三章末亦建立此問題之雙目標規劃模式，並以此模式所求結果驗證 Multi-TagSNP 演算法的正確性。第四章以拉氏鬆弛法求解 tagSNP 選取問題，修正一般拉氏鬆弛法之演算流程以加快求解效率與提升求解品質，並執行數值分析比對我們所提出之演算法與最佳化軟體 CPLEX 之求解品質與效率；第五章歸納本論文，最後提出一些後續研究之方向，其中我們特別針對如何在具有容量限制的生物晶片上選取一組 tagSNP 使其在應用於辨識上最為可靠提出一個整數規劃模式，呈現其最佳容量限制與辨識次數下限（可視為可靠度或強健性）之關係圖，以幫助業者決定如何選取適當的 tagSNP 以植入生物晶片中。

第二章 文獻回顧

本章主要對 tagSNP 選取問題之相關文獻與所應用之生物背景資料作整理回顧。2.1 小節介紹如何將生物觀念引入研究問題或方法，並統整相關文獻中 Block 與 tagSNP 之不同定義。2.2 小節描述一些不同模式的 tagSNP 選取問題，並整理一些求解方法與結果。2.3 小節以組合數學的角度觀看 tagSNP 選取問題，解釋爲何 tagSNP 選取問題亦為一個集合涵蓋問題，並說明何其理論複雜度。由於 tagSNP 選取問題可視為一種集合涵蓋問題，因此最後在 2.4 小節介紹集合涵蓋問題的模式與過去文獻所使用的求解方法。

2.1 tagSNP 選取問題之生物資訊相關背景

2.1.1 限制 Diversity 與 Block 結構

在基因的演化過程中，重組（recombination）在影響個體差異方面扮演著重要角色，透過重組可以提升物種的多樣性。學者研究發現在基因序列中並非所有基因座發生重組的機率皆相同，有些基因座發生重組的機率極高，有些幾乎不曾發生過(Phillips et al.，2003)；重組頻率高的位置稱為 hot spot，而 hot spot 的出現將限制基因序列的差異性（Diversity)。以圖 2.1 為例，假設兩條相異之 DNA 序列發生重組配對（此處為便於觀察分別以白色及灰色代表原各自之 DNA 序列)，倘若重組僅發生在 hot spot 上，則圖中的 DNA 序列在發生重組後只可能產生八種不同的組合情形；倘若沒有 hot spot 的影響，則每一點（共 12 個）皆可能發生重組，如此共可能產生 2^{12} 種不同的組合情形。由此可知 hot spot 會大大限制了基因序列的 Diversity，因而造成序列產生如圖中區塊（Block）的結構（Wang et al.，2002；Arnheim et al.，2003)。

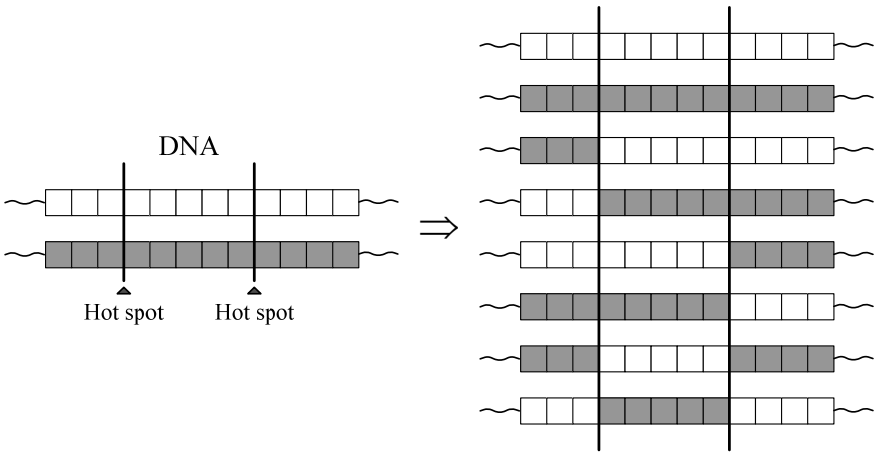


圖 2.1：DNA 序列之 Block structure

由於 DNA 序列確實具有 Block 結構（Gabriel et al.，2002），且不同 Block 中序列之 Diversity 相異不大，因此有學者提出以固定 Diversity 值來切割 Block 的方法，而不同的 Haplotype diversity 計算方式將導致不同的 Block 切割結果。為了更了解以 Diversity 作為切割依據的運作模式，以下先介紹一種原理簡單且為多數人使用的 Diversity 計算方式，接著再敘述如何將之用於切割 Block 上。

由圖 2.1 中可觀察出 DNA 序列中相同區塊位置即成為一個 Block，而 Diversity 的大小由 Block 中不同的 Haplotype 樣式決定，因此可將 Diversity 用簡單的數學式表示：$D=1-\sum_{i=1}^{m} H_i^2$，其中 H_i 表示 Block 中不同 Haplotype 樣式所占的比例（Li and Guaur，1990），以下以一小範例說明：

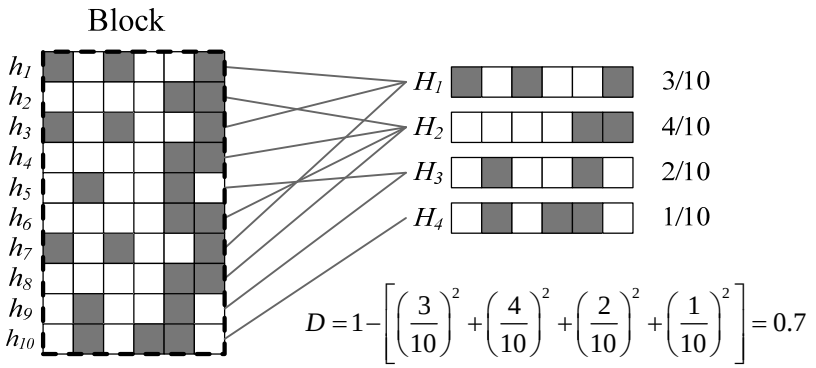


圖 2.2：Diversity 之計算

將圖 2.2 左邊的 Block 中所有的 Haplotype 依樣式分類，其分類結果如圖 2.2 右半部所示；共有四種不同的 Haplotype 樣式， 而各 Haplotype 樣式在 Block 中出現的頻率被標示於其右邊，接著依照公式即可計算出 Diversity 值為 0.7。

使用 Diversity 切割 Block 時，會先訂定一個可接受的門檻值，接著逐步在 Block 中增加 Column 並計算增加 Column 之後的 Diversity 值，以使 Block 的 Diversity 值逐漸逼近門檻值，直到再增加某一 Column 會導致其 Diversity 值超過門檻值時為止（此時不會將該 Column 加入）。如圖 2.3 所示，假設使用者訂定之門檻值為 0.68，將圖 2.2 Block 中的 Haplotype 重新切割，依照上述方式逐步增加 Column 並計算 Diversity 值於下方，由圖 2.3 可知當增加到四個 Columns 時，Diversity 已超過門檻值 0.68，因此最後切割之 Block 為三個 Columns。

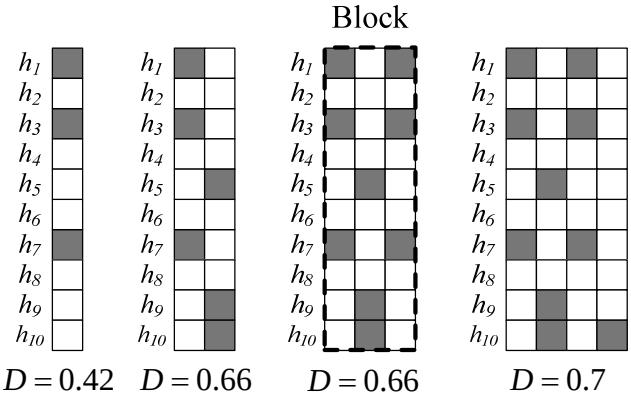


圖 2.3：設定 Diversity 門檻值切割 Block

2.1.2 連鎖不平衡

在 DNA 序列中，任兩個基因座上的等位基因彼此間可能具某種關連程度而非完全隨機分配，此種現象稱之為等位元基因關聯（allelic association）或稱為連鎖不平衡（Linkage Disequilibrium，LD）。就統計學的觀點來看，假若某群體在初始狀態時

具有連鎖不平衡之情形，其在隨機配對 n 代之後，將會使該連鎖不平衡狀態隨著代數增加而逐漸減少。在基因重組率（recombination rate）很大（≈ 0.5）時，連鎖不平衡將會隨著代數的增加而迅速減小；反之，當兩個基因座相連時，其重組率很小（接近 0），而不平衡狀態亦將持續很多代。由於探究連鎖分析必須要有多代的群組資料來測試，在資料收集上困難度頗高，因此過去研究中通常以基因座的相連程度作為 LD 值計算方法之依據。

在計算 LD 值的相關文獻中，有許多統計學家模擬重組配對過程而提出多種計算 LD 值之方法（Devlin and Risch，1995），其中最常被使用的兩類方法分別是（2.1）式中用以計算兩個 loci 間對偶基因的關係係數 r^2（其數值介於 0～1），以及（2.2）式中的標準化連鎖不平衡參數 D'（其數值介於 -1～1）：

$$r^2 = \frac{\left(P_{AB} - P_A P_B\right)^2}{P_A\left(1-P_A\right)P_B\left(1-P_B\right)} \qquad (2.1)$$

$$D = P_{AB} - P_A P_B$$
$$D'_{A,B} = \begin{cases} \dfrac{D}{\min\left(P_A \times P_B, P_a \times P_b\right)}, \ if\ D < 0 \\ \dfrac{D}{\min\left(P_A \times P_b, P_a \times P_B\right)}, \ if\ D > 0 \end{cases} \qquad (2.2)$$

其中 P_A、P_B 分別表示在 SNP S_A 與 S_B 各自的 major allele（主要對偶基因，為母體中基因座上出現機率較大者）之出現頻率；而 P_a、P_b 表示 S_A 與 S_B 上各自的 minor allele（次要對偶基因，為母體中基因座上出現機率較小者）之出現頻率（因此 $P_a = 1 - P_A$；$P_b = 1 - P_B$）；P_{AB} 為 S_A 與 S_B 同時出現其 major allele 的頻率。舉例來說，對圖 3.1 的矩陣 H 而言，由於 S_1 中 1 的出現次數多於 0（假設母體狀況亦同），因此 S_1 的 major allele 為 1（而其 minor allele 則為 0），令 $P_{S_j^{\text{major}}}$（or $P_{S_j^{\text{minor}}}$）代表 S_1 的 major（or minor）allele 的出現頻率，則 $P_{S_1^{\text{major}}} = 3/4 = 0.75$，$P_{S_1^{\text{minor}}} = 1/4 = 0.25$；另外，當 major allele 與 minor allele 出現頻率相同時，我們可任意訂定其 major（or minor）allele 為 1 或 0。舉例來說，S_3

的 0 與 1 出現次數相同，因此 $P_{S_3^{\text{major}}} = P_{S_3^{\text{minor}}} = 0.5$ 而其 major（or minor）allele 可為 1 或 0。假設我們令 S_3 的 major allele 為 0，則 $(1,0)$ 即為 S_1 與 S_3 同時出現 major allele 的組合，且其在 S_1 與 S_3 兩個 column 中總共出現 1 次，因此 $P_{S_1^{major}S_3^{major}} = 1/4 = 0.25$。由以上分析，我們可依式子（2.1）（2.2）計算出兩種估算 S_1 與 S_3 間的 LD 值如下：

$$r^2_{S_1,S_3} = \frac{(0.25 - 0.75\times 0.5)^2}{0.75\times 0.25\times 0.5\times 0.5} \approx 0.3333$$
$$D = 0.25 - 0.75\times 0.5 = -0.125 < 0$$
$$D'_{S_1S_3} = \frac{-0.125}{\min(0.75\times 0.5, 0.25\times 0.5)} = -1$$

其中，若 r^2 與 $|D'|$ 越趨近於 1，則表示所分析的兩個 SNP 間的關連性越高。由於 $|D'|$ 將 minor allele 亦列入考慮，因此對演化過程中其基因座上是否發生重組現象較為敏感，當 $|D'| = 1$ 時稱為 complete linkage disequilibrium，代表此兩個基因座過去並未因重組現象而被分開。值得注意的是，Lin（2005）的研究指出在較少的資料量情況時，用 $|D'|$ 來估算 LD 值容易與實際造成較大的誤差。至於整個矩陣 H 的 LD 值估算方式，則必須先算出該矩陣所有的兩兩 SNP 間之 LD 值，再進一步計算這些 LD 值的平均以作為該矩陣之 LD 估算值。

2.1.3 Haplotype Block 之定義

由於每一個 Haplotype 可能含有幾百萬個 SNP，我們可預期在如此大規模的 SNP 中篩選 tagSNP 將會十分困難。為了解決此一困難，亦有多位學者提出分段求取 tagSNP 的方式。該類方式會將一個長的 Haplotype 序列切割成多個較短的 Haplotype Block，再分別對每個 Block 做 tagSNP 的選取。且由 2.1 小節敘述得知，基因序列確實具有 Block 結構，因此採用 Block 的方式縮小 tagSNP 選取問題之規模的確具其可行性。歸納過去學者對於此問題之研究，主要可將 Haplotype Block 分成三種定義：

1. Patil et al.（2001）提出以限制 Haplotype Diversity 作為切割的準則，訂定需求 Diversity 之門檻值，以使至少有某一比例的 Haplotype（在母體上稱之為 Common Haplotype）必須為受測個體所共有；這些學者通常會以 80%的差異度作為切割 Haplotype Block 的門檻值依據。
2. Gabriel et al.（2002）提出 LD-based Blocks，運用連鎖不平衡的觀念將 LD 值高過所訂定門檻值之相鄰 SNP pair 歸類於同一個 Block。就生物上的意義來看，當 LD 值高時，表示此區的鹼基不易發生配對而改變；而當 LD 值低時，表示此區為重組配對的 hot spot；因此，依此方式做 Block 區隔可降低在配對時的 Diversity，亦具有生物意義。
3. Wang et al.（2002）根據歷史資料將尚未發生四種配偶子（gamete）重組之 SNPs 連續序列定義為一個 Block。

2.1.4 tagSNP 之定義

近年來有不少學者致力於挑選tagSNP之研究上，根據現有狀況及不同的使用目的而延伸出多種不同的tagSNP定義，以下統整三種較多人沿用的定義：

1. 可辨識所有Haplotype樣式的tagSNP

以極小化所選取的tagSNP個數為目標，將tagSNP定義為足以辨識所有受驗者之Haplotype樣式的SNP集合。由於實務上可能僅需辨識某一使用者需要的精確度，而不需完整辨識所有不同的Haplotype樣式，因此可進一步讓使用者自行調整辨識程度以改善處理效率。在這方面的相關研究中，一般的作法通常先探討可完整辨識出所有Haplotype的SNP，接著再放寬辨識程度至可接受的誤差下加快求解速率。在先前文獻中，已有學者Patil et al.（2001）、Zhang et al.（2002）、Avi-Itzhak et al.（2003）分別採用Greedy Algorithm、Dynamic Programming Algorithm、Simple Numerical Algorithm等方式來求解此類tagSNP選取問題。

2. 符合Diversity門檻值之tagSNP

Johnson et al.（2001）將tagSNP定義為一個必須含有最少之元素個數且其所構成之Haplotype Diversity亦將逼近既定門檻值之SNP子集合。其作法先將所有可能的SNP皆納入考量，計算被選取之SNP所構成的Haplotype Diversity，最後選取Diversity值超過門檻值且元素個數最少之SNP子集合。

3. 以類似之LD值群組（LD-bin）涵蓋完整Haplotype序列資訊之tagSNP

以LD值為選取概念，對單一SNP而言，將所有其他SNP與此SNP間具有高LD值者設為同一群組（表示擁有之資訊相似），每個SNP群組中僅取其一以作為該群組之代表；最後選取部分SNP集合（稱為LD-bin），使其可涵蓋所有Haplotype資訊。LD-bin中的SNP不一定緊密相連，可能原先散落於Haplotype序列各處，此點與Block之觀念（SNP為連續序列）較有差異。文獻中，Carlson et al.（2004）、Wang et al.（2002）等學者將LD-bin視為所挑選之tagSNP集合；Carlson et al.（2004）以任兩個SNP為一組配對（pair），計算所有配對組的LD值（亦可將此值視為correlation；r^2）以作為SNP的預測能力；對單一尚未被挑選的SNP，計算其與其他SNP間之r^2，將其所有的r^2值中最大者視為此SNP之預測能力；而對於所有尚未被挑選的SNP群，選取其r^2最小者作為整體SNP群的預測能力。依此定義，tagSNP即為具有最少的SNP個數且其整體預測能力（即LD值）超過預設之門檻值的SNP群組。

2.2 tagSNP選取問題

由於文獻中的tagSNP具有多類不同定義，且採用Block切割Haplotype序列與否亦會影響tagSNP的選取問題模式，本小節將整理文獻中各類的tagSNP選取問題模式以及其求解方法。

2.2.1 tagSNP選取問題模式

Bafna et al.（2003）在沒有切割Block的情況下，以極大化可擷取的Haplotype資訊為目標來求取tagSNP；相同情況下，Ke and Cardin（2003）則以最小化tagSNP數為目標切入。而在考量切割Block情況下，Patil et al.（2001）、Zhang et al.（2002）以限制Diversity的方式切割Block，在每一Block中求取最小化tagSNP之數目。

另一種tagSNP選取問題則是先限制tagSNP數目（Zhang and Jin，2003），再依此限制求取Block的切割方式以涵蓋最大範圍的Haplotype序列；Zhang et al.（2002a）、Weale et al.（2003）等學者則在給定tagSNP數目下，求解最大化Haplotype資訊的tagSNP。除此之外，部分研究者將重點擺在使用tagSNP預測整體Haplotype序列，如Halperin et al.（2005）在給定tagSNP數目下，選取tagSNP以使預測Haplotype序列之期望誤差最少。在這些相關研究議題中，有的文獻將重點擺放在切割Block的方法上，有的將重點放於挑選tagSNP之方法上，有的則是兩者並重；然而這些文獻之最終目的皆在於縮小原SNP序列資料，使之應用於實務上可獲得較高之執行效率。

2.2.2 tagSNP選取問題之求解方法與結果

在 tagSNP 的求解方法上，根據不同 tagSNP 定義之各類 tagSNP 選取問題已有許多學者提出多種不同的演算方法。Avi-Itzhak et al.（2003）採用列舉的方式（simple numerical algorithm）求解 Block size 較小之 tagSNP 選取問題，該研究分別模擬選取 45 個非洲人與高加索人之 6 號、21 號，以及 22 號染色體上的 tagSNP，並提出修正之演算法以處理不完全辨識 Haplotype 差異之 tagSNP 選取問題，以及模擬兩種情況下（是否完全辨識 Haplotype 差異）的求解結果。在考慮完整辨識的情況下，非洲人可節省 25%的 SNP，高加索人則可節省 36%之 SNP；另外，若在允許遺漏 10%的辨識資訊情況下，所需的 SNP 數更少，非洲人可節省 38%的 SNP，高加索人則可節省全部的 49%。

Patial et al.（2001）先使用 Greedy 演算法找出切割 Block 之分界線，使其滿足每一個 Block 中至少有 80%的 Haplotype 重複出現(該類 Haplotype 被定義為 Common Haplotype)；在此種 Block 條件下求取最少個數之 tagSNP，此 tagSNP 可辨識每一個 Block 中 80%的 Haplotype 樣式;其結果將第 21 號染色體中 24047 個 SNP 切割成 4135 個 Block，並從中選取出 4563 個 tagSNP；Zhang et al.（2002）採用 Patial et al.（2001）的 Block 定義，用動態規劃（Dynamic Programming Algorithm）切割 Block ，並在其中求取最少個數之 tagSNP；其結果將第 21 號染色體切割成 2575 個 Block 並從中選取出 3562 個 tagSNP。從上述的結果顯示兩種演算法的效率皆很高，且亦可從結果推論出，不同的 Block 切割方式也會影響到 tagSNP 的選取結果。

Huang et al.（2005）提出兩種 Greedy 演算法與一種反覆式線性鬆弛法以求解在考慮遺漏部分資訊的情況下的 tagSNP 選取問題。該研究模擬 4 種不同出處的資料，比較其兩種 Greedy 演算法與反覆式線性鬆弛法的求解品質與效率。其結果顯示，反覆式線性鬆弛法求解品質最佳，而另兩種 Greedy 演算法的求解品質與效率亦不錯，且在考慮遺漏部分資訊的情況下，第二種 Greedy 演算法又比第一種 Greedy 演算法有效率。

Carlson et al.（2004）亦提出一個 Greedy 演算法，以極大化可獲得之資訊來選取 tagSNP。該研究模擬了 47 個獨立個體（24 個非裔美國人與 23 個歐洲人）的 100 個基因，總共有 8,877 個 SNP，於非裔美國人中選取出 3,178 個 SNP 作為 tagSNP；歐洲人則選取出 2,375 個 SNP 作為 tagSNP；雖然此篇研究主要重點在於闡述其結果數據的生物意義，而非著重於改良其演算法之求解品質與效率，然而若由其最終所選取之 tagSNP 的縮減模看來，我們亦可推論出該演算法的求解品質其實並不差。

2.3 tagSNP選取問題之理論複雜度

從 2.1.4 小節可得知 tagSNP 有許多不同的定義，但是不論在哪一種定義下，若我們將已選取的 SNP 所代表之資訊視為一個資訊集合，則 tagSNP 選取問題即為一個旨在選取最少個資訊集合以獲得全部 Haplotype 資訊之問題，如此一來 tagSNP 選取問題即可被視為一個集合涵蓋問題（Set Covering Problem，SCP）。由於 SCP 是一個 NP complete 問題（Garey and Johnson, 1979），因此可推論 tagSNP 選取問題亦為一 NP complete 問題，無法在多項式時間內求得最佳解。

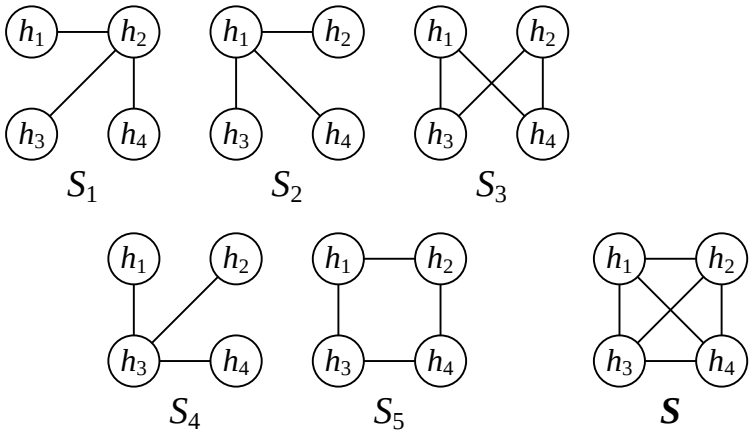
若近一步分析 tagSNP 選取問題，將 Haplotype 視為待測之個體項目（item），不同的 SNP 分別表示不同屬性的測驗（test），我們可以將 tagSNP 選取問題視為一個用最少個 test 以辨識所有不同 item 之問題。由於任一個體均可用不同屬性之集合來辨識出來，因此任兩個體間必定有一個屬性可以辨識他們之間的差異，此類問題稱之為 Test Covering Problem（TCP）。Garey and Johnson（1979）証明 TCP 為一 NP-hard 問題，理論上仍無法設計出多項式時間之最佳解演算法。

將每一個屬性測驗以個體間之辨識關係圖表示，若可以辨識則將兩個體連結；以 tagSNP 選取問題來看，視每一可選取之 SNP 為一屬性測驗（test），h_i 為待測之個體（item），以圖 2.4（a）之矩陣樣本為例，以 S_1 欄位來看，其中 h_1 對應之列值為 1，h_2 對應之列值為 0，表示對 S_1 可辨識 h_1 與 h_2；反之，h_1 對應之列值為 1，h_3 對應之列值亦為 1，表示對 S_1 不能辨識 h_1 與 h_3。將上述關係圖 2.4（b）表示，譬如 S_1 可以辨識 h_1、h_2，因此將 node h_1 與 node h_2 兩節點連結起來（arc_{h_1,h_2}），同理可連結 arc_{h_2,h_3}、arc_{h_3,h_4}，這些節點節線可構成一個 S_1 的辨識關係圖。以此類推，可將所有的 S_j 分別以 j 個圖示表示，如此可將 tagSNP 選取問題視為以最少個辨識關係圖（S_j）使其圖形的聯集可成為一個完整圖（complete graph），亦即一個 SCP；換個角度來看，本問題亦可被視為如何自一個完整圖扣除最少相同的關係圖（S_j），亦即將每個 S_j 視為 Cut，

而使所有的節線全部被刪除的 Cut Covering Problem。

$$H = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} S_1 & S_2 & S_3 & S_4 & S_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

(a) Haplotype之矩陣範例H



(b) 矩陣H之辨識關係圖

圖 2.4：tagSNP 選取問題與 TCP 之關聯

2.4 集合涵蓋問題

由 2.3 小節可知 tagSNP 選取問題本質上為一個集合涵蓋問題，因此求解 tagSNP 選取問題相當於求解一個 SCP。因此本小節主要介紹 SCP 的問題模式，以及統整求解該問題之相關文獻。

2.4.1 問題模式

SCP被應用於多種實務上，例如：排班問題、車輛路線問題、異質型無線感測網路上的目標點涵蓋問題等，許多相關的求解演算法陸續被提出來。其基本架構如

P_{SCP} 模式，其中 $A=\left[a_{i,j}\right]$ 包含了 M 個列（row）向量（ $M=\{1,\ldots,m\}$ ）以及 N 個行（column）向量（ $N=\{1,\ldots,n\}$ ），為一個 $m\times n$ 的0,1矩陣。 $C=\left[c_j\right]$ 為一個 n 維的向量，且 $c_j, \forall j\in N$ 代表第 j 行之成本，對每一行成本而言，假設所有成本值皆大於0。由於集合涵蓋問題目的在於利用不同集合的聯集涵蓋所有不同項目，因此可知每個項目至少需被涵蓋一次，將每一行視為不同之集合，每一列視為需被涵蓋的不同項目，如此可得每一列（ $i\in M$ ）至少需被一行（ $j\in N$ ）涵蓋到，以 $a_{i,j}=1$ 表示第 j 行涵蓋第 i 列， $a_{i,j}=0$ 表示第 j 行未能涵蓋第 i 列。SCP之目的為在符合涵蓋所有項目的限制下，求取所花費總成本最小之集合聯集。

$$\begin{aligned} \min\ & \sum_{j\in N} c_j x_j \\ st\quad & \sum_{j\in N} a_{ij} x_j \geq 1,\ \ i\in M \\ & x_j\in\{0,1\},\ \ j\in N \end{aligned} \qquad (P_{SCP})$$

2.4.2 求解方法

在求解集合涵蓋問題時，一般常使用上下界夾擠的方法來尋找精確解，可先設法求得一組可行解作為精確解之上界，下界值一般透過求解放鬆限制式之問題而得。最常被採用之放鬆限制式方法為拉氏鬆弛法（Largrangian Relaxation），此方法保持原變數之限制條件，也就是變數仍為binary變數，透過放鬆整條限制式的方式，將限制式移至目標式，給定一違背此限制式之懲罰成本變數，將該變數稱之為拉氏乘數，一旦違背某限制式，便給予目標式一懲罰值。此種拉氏鬆弛法的求解方式通常採用次梯度法（Subgradient），透過逐漸更新拉氏乘數獲得最佳拉氏乘數向量，為一種極具效率的啟發式演算，相關文獻如Ceria et al.（1998）、Haddadi（1997）與Caprara et al.（1999）皆採用啟發式拉氏鬆弛法求解集合涵蓋問題。

另一種放鬆方法為線性鬆弛法（Linear Relaxation），將整數限制式放鬆，也就

是將binary變數改為介於0~1的線性變數，但不變動目標式與其他限制式。求解此種放鬆問題之方法可採用行運算技巧變數產生法（Column Generation，CG），最早被Gimore及Gomory（1961；1963）使用在求解Cutting Stock Problem之演算法流程中；此方法在集合涵蓋問題的使用上常出現在當集合變數不易完全獲得時，因CG法不需考慮所有可能的集合變數，此種方法又常與Dantzig-Wolfe分解演算法合用，利用對偶理論（Dual Theory）來產生變數（column），且僅考慮部分變數而非所有變數，如此可加快求解速率，透過子問題與主問題的搭配求解過程中，逐漸收斂獲得最佳解。

由於 SCP 被證實是一個 NP complete 問題（Garey and Johnson，1979），無法在多項式時間內求得最佳解，因此多種啟發式方法被提出來，常用的啟發式演算法如貪婪演算法（Chvatal，1979；Slavík，1996），此種演算法通常是在每一遞迴步驟中選取對目標式最有利之集合，或是選取某集合替換已被選取之集合中的子集合，使其可改善最多目標值。Feo et al.（1989）提出一個類似 Chvatal（1979）演算法的通式，改變挑選規則，使之不是單純自 degree 最大的集合開始挑選，而改用一種隨機的概念自大於最大 degree 值的某一比例開始挑選即可。除此之外，尚有其他演算法被提出，如 Hifi（2000）、Ohlsson et al.（2001）以類神經網路法求解；Beasley and Chu（1996）、Al-Sultan et al.（1996）則採用基因演算法求解等。

2.5 小結

本章節介紹 tagSNP 選取問題的種類以及所應用到的生物概念，從文獻回顧中可發現並無學者針對相同選取個數，不同選取結果（多重最佳解）做比較，因此本論文之第三章將以生物觀念重新檢視求得之多重解最佳解，將 2.1.2 小節中的 LD 值大小視為任兩 SNP 間的關連程度，由於文獻中亦有學者用 LD 值作為 Haplotype 序列資訊量的評估（Bafna et al., 2003；Carlson et al.,2004），因此本論文亦將此觀念加入演算

過程，作為選取最後推薦解的第二種評選準則。文獻回顧最後歸納了集合涵概問題的相關文獻，從文獻中發現拉氏鬆弛法用於求解大規模的集合涵概問題十分有效率，因此本論文的第四章採用拉氏鬆弛法的概念發展一個啟發式演算法，其主要目的在提升求解大規模的 tagSNP 選取問題的效率。

第三章 求解 tagSNP 選取問題之多重最佳解

此章節首先將 tagSNP 選取問題轉換為數學模式 P_{tagSNP}（詳見 3.1 小節），提出一個用圖形理論為基礎之 Multi-TagSNP 演算法以求得 P_{tagSNP} 的多重最佳解，並以 LD 值之和作為評選準則，在所找出之多重最佳解中再進一步選出一個最佳的推薦解。3.2 小節中描述 Multi-TagSNP 演算法的各個步驟，其各步驟之時間複雜度將於 3.3 小節加以估計分析，而 3.4 小節亦將以一則範例演練演算法之各步驟。最後，我們於 3.5 小節提出一個雙目標整數規劃模式 P^{M}_{tagSNP}，並將 3.4 小節中的範例以雙目標規劃方式求解，用以說明 Multi-TagSNP 之求解過程。

3.1 tagSNP 選取問題之數學模式

若以數學的角度來看 tagSNP 選取問題，由於單點發生突變的機率很低，可以推論同一點發生兩次突變的可能性十分渺小，因此學者通常會假設單點上僅可能發生一次突變（$0 \rightarrow 1$），故我們可將 Haplotype 的 SNP 鹼基用 0, 1 表示，標記為 0 代表野生型（wild type），標記為 1 代表突變型（mutant type）。一列 Haplotype 是由多個 SNP 鹼基組成之序列，因此 m 個 Haplotype $\{h_i : i = 1,...,m\}$ 內含 n 個 SNP $\{S_j : j = 1,...,n\}$ 可表示為一個 $m \times n$ 的 0,1 資料矩陣，如圖 3.1 矩陣 $H = [H_{i,j}]$ 所示。

為了比較 Haplotype 間之差別，我們可將任兩個 Haplotype h_{i_1} 與 h_{i_2} 配對成 $E_{i_1 i_2} = (h_{i_1}, h_{i_2})$，而令 $E = \{E_{i_1,i_2} : i_1 = 1,...,m; i_2 = 1,...,m\}$ 代表所有可能的 Haplotype 配對所構成之集合。舉例來說，h_1 與 h_2 可配對形成 $E_{1,2}$；h_1 與 h_3 可配對形成 $E_{1,3}$。圖 3.1 有四個 Haplotype，總共形成 $C_2^4 = 6$ 種配對方式，因此集合 $E = \{E_{1,2}, E_{1,3}, E_{1,4}, E_{2,3}, E_{2,4}, E_{3,4}\}$。另外，針對每一個 S_j 及 E_{i_1,i_2} 我們可定義一個 indicator function $I(E_{i_1,i_2}, S_j) \in \{0,1\}$，其中若 $(h_{i_1,j}, h_{i_2,j}) \in \{(0,0),(1,1)\}$ 時，S_j 將無法被用來辨識該組 Haplotype pair E_{i_1,i_2}，因此

$I(E_{i_1,i_2},S_j)=0$；相反地，假若 $(h_{i_1,j},h_{i_2,j})\in\{(0,1),(1,0)\}$ 時，S_j 可被用來辨識該組 Haplotype pair E_{i_1,i_2}，而 $I(E_{i_1,i_2},S_j)=1$。如圖 3.1 以 h_1、h_2 與 S_3 之鹼基交集位置為例，由於 $(h_{1,3},h_{2,3})=(0,0)$，代表 S_3 不能被用來辨識 $E_{1,2}$，因此 $I(E_{1,2},S_3)=0$；同理，由於 $(h_{3,3},h_{4,3})=(1,1)$，代表 S_3 不能被用來辨識 $E_{3,4}$，因此 $I(E_{3,4},S_3)=0$；反之，$(h_{2,3},h_{3,3})=(0,1)$，代表 S_3 可被用來辨識 $E_{2,3}$，因此 $I(E_{2,3},S_3)=1$。以此類推，可得 S_3 可否被用來辨識 E 中所有 Haplotype pair 的狀態，如圖 3.1 右半部所示。

$$H=\begin{array}{c} \\ h_1\\h_2\\h_3\\h_4\end{array}\begin{array}{c}\begin{array}{ccccc}S_1&S_2&S_3&S_4&S_5\end{array}\\\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\1&1&1&0&0\\1&1&1&1&1\end{bmatrix}\end{array}\Rightarrow\begin{array}{c}S_3\\\begin{bmatrix}0\\0\\1\\1\end{bmatrix}\end{array}\begin{array}{l}\\E_{1,2}\\E_{2,3}\\E_{3,4}\end{array}\begin{array}{l}E_{1,3}\\E_{2,4}\\E_{1,4}\end{array}\Rightarrow$$

$$\begin{aligned}&(h_{1,3},h_{2,3})=(0,0)\Rightarrow I(E_{1,2},S_3)=0\\&(h_{1,3},h_{3,3})=(0,1)\Rightarrow I(E_{1,3},S_3)=1\\&(h_{1,3},h_{4,3})=(0,1)\Rightarrow I(E_{1,4},S_3)=1\\&(h_{2,3},h_{3,3})=(0,1)\Rightarrow I(E_{2,3},S_3)=1\\&(h_{2,3},h_{4,3})=(0,1)\Rightarrow I(E_{2,4},S_3)=1\\&(h_{3,3},h_{4,3})=(1,1)\Rightarrow I(E_{3,4},S_3)=0\end{aligned}$$

圖 3.1：SNP 及其所能辨識之 Haplotype pair 範例

針對所有的 SNP（S_j）重複其與所有 E_{i_1,i_2} 的比對步驟，可建構一個 Haplotype pair 與 SNP 間的比對矩陣 ES，該矩陣內的元素即為其所對應的 $I(E_{i_1,i_2},S_j)$ 之值，如圖 3.2 之矩陣 H 經轉換後可得比對矩陣 ES。

$$H=\begin{array}{c} \\ h_1\\h_2\\h_3\\h_4\end{array}\begin{array}{c}\begin{array}{ccccc}S_1&S_2&S_3&S_4&S_5\end{array}\\\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\1&1&1&0&0\\1&1&1&1&1\end{bmatrix}\end{array}\quad\Rightarrow\quad ES=\begin{array}{c} \\ E_{1,2}\\E_{1,3}\\E_{1,4}\\E_{2,3}\\E_{2,4}\\E_{3,4}\end{array}\begin{array}{c}\begin{array}{ccccc}S_1&S_2&S_3&S_4&S_5\end{array}\\\begin{bmatrix}1&1&0&0&1\\0&1&1&1&1\\0&1&1&0&0\\1&0&1&1&0\\1&0&1&0&1\\0&0&0&1&1\end{bmatrix}\end{array}$$

圖 3.2：比較 Haplotype pair 差異之轉換矩陣

若將 *ES* 視為一個 bipartite 網路圖 $G=(N,A)$ 的 adjacency matrix，我們可將所有的 SNP（S_j）與 Haplotype pair（E_{i_1,i_2}）視為 S 節點與 E 節點，亦即 $N=S\cup E$；針對所有 $I(E_{i_1,i_2},S_j)=1$ 之關係，我們可將其所對應之 S、E 節點連結，亦即 $A=\{(S_j,E_{i_1,i_2}):\forall S_j\in S,\ E_{i_1,i_2}\in E\}$；反之，對所有 $I(E_{i_1,i_2},S_j)=0$ 之關係，其所對應之 S、E 節點將不被連結。如此一來即可將 SNP 與 Haplotype pair 的比對關係以一個 bipartite 網路圖 $G=(N,A)$ 表示，並將各節點 $k\in N$ 的 degree（deg(k)）記錄於其旁。舉例來說，圖 3.3 即為圖 3.2 的 *ES* 矩陣所對應之 bipartite 網路圖 $G=(N,A)$。

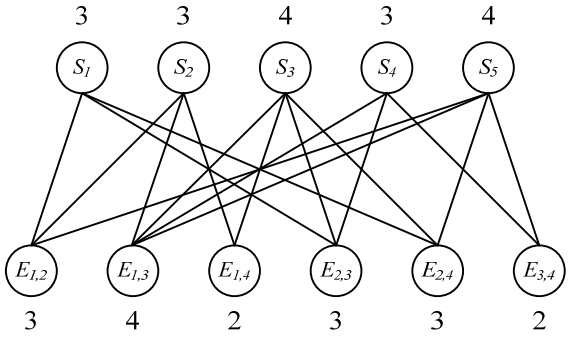


圖 3.3：說明 SNP 與 Haplotype pair 辨識關係之 bipartite 網路圖

由圖 3.3 可觀察出，當 S_1 被挑選時，$E_{1,2}$、$E_{2,3}$、$E_{2,4}$ 皆可被其所辨識。選取 tagSNP 之主要限制在於所選出之 SNP 集合必須保證可以辨識 *E* 集合內的所有元素；此外，在所有滿足此限制條件的候選 SNP 集合中，tagSNP 所包含的 SNP 個數必須為最少。以圖 3.4 為例，選擇$\{S_1,S_2,S_4\}$可以辨識出所有的 *E* 元素；同理，選擇$\{S_3,S_5\}$也可以辨識出所有的 *E* 元素；由於$\{S_3,S_5\}$的個數為所有可辨識 *E* 集合全部元素之 SNP 集合中最小者，因此$\{S_3,S_5\}$將被選為 tagSNP。值得注意的是，在 tagSNP 的選取問題中，其最佳的 tagSNP 解可能不只一個。

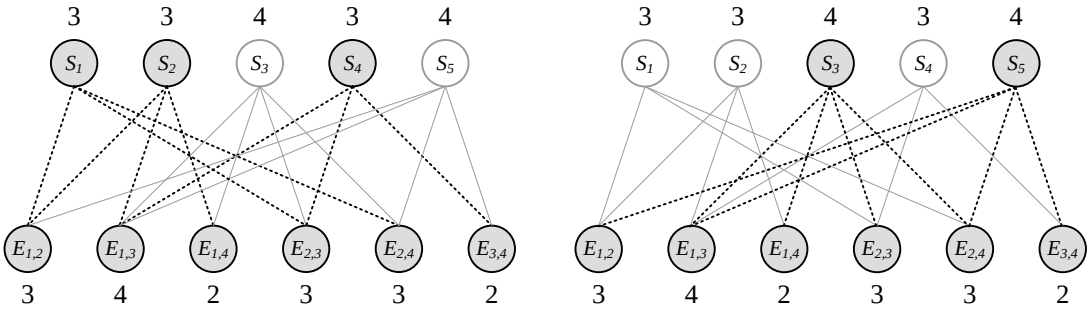


圖 3.4：挑選 tagSNP 範例圖

以下我們將針對 tagSNP 的選取問題依上述規則建立成數學模式：假設 x_j 代表 S_j 是否被挑選為 tagSNP 的 0,1 決策變數，其中 $x_j = 1$ 表示 S_j 被選取為 tagSNP；反之，$x_j = 0$ 表示 S_j 未被選取為 tagSNP。針對每個 Haplotype pair E_{i_1,i_2} 而言，應該至少會存在某個 S_j 可使 $I(E_{i_1,i_2}, S_j) = 1$，因此 tagSNP 選取問題可用以下模式 P_{tagSNP} 代表之：

$$
\begin{aligned}
&Min \quad \sum\nolimits_{S_j \in \mathrm{S}} x_j && (\mathrm{P}_{tagSNP}) \\
&st. \quad \sum\nolimits_{(S_j, E_{i_1,i_2}) \in A} x_j \geq 1, \ \forall E_{i_1,i_2} \in \mathrm{E} \\
&\qquad x_j \in \{0,1\}, \ \forall S_j \in \mathrm{S}
\end{aligned}
$$

P_{tagSNP} 通常存在多重最佳解，而過去學者在求解此問題時，大都僅止於求得一個最佳解，而將研究重點放在 tagSNP 之選取方法及其求解效率，並未深入去探討哪個最佳解最能代表其原始資料所表達的資訊。由第二章的文獻中可得知基因序列中存在連鎖不平衡現象，加上許多學者亦將 LD 值作為評估 Haplotype 序列的資訊量（Bafna et al., 2003；Carlson et al., 2004），因此本研究亦將引進 LD 值作為評選多重最佳解中關聯性高低之準則，對於所求得之多重最佳解加以計算其個別最佳解的 LD 值之和，也就是對已選取之 tagSNP 中的每一元素計算該元素與任一未被選取的 SNP 彼此間的 LD 值，將每組 tagSNP 中各元素所計算的 LD 值加總取平均值做為此組解的 LD 值表現，該 LD 值代表該組 tagSNP 所能擷取剩餘未挑選的 SNP 之序列資訊量。最後，從

所有多重最佳解中選出其 LD 值之和最高者以作為最終的推薦解；該推薦解不僅具有最少個數的 tagSNP，且可涵蓋最多的 SNP 資訊，因此更能代表整個 Haplotype 序列，以期未來不僅能辨識 Haplotype 樣式，亦能更進一步推論出極為近似的 Haplotype 以供使用者作其它研究使用。

3.2 Multi-TagSNP 演算法

文獻中所提出的 tagSNP 選取方法皆只能求出一個最佳解，而對於求取其它的多重最佳解並無加以著墨。本研究擬提出一個啟發式的圖形演算法（Graph Algorithm）求出所有的多重最佳解，並針對這些多重最佳解加以進一步處理其解當中所隱含的 LD 資訊，以得出更有用的 tagSNP。我們將此演算法命名為 Multi-TagSNP，其步驟如下所示：

Step 1：刪除矩陣 H 中重複及互補的 column（亦即刪除相同的 column 以及數值完全相反的 column），並記錄所刪除的 column 群組，令經過處理之後的矩陣為 H'。

Step 2：針對新的矩陣 H' 建立 bipartite 網路圖 $G=(N,A)$，形成 S 節點群與 E 節點群；將 E 節點中 degree 為 1 者其對應之 S 節點放入 tagSNP 集合中，並刪除此節點與其所連結之相關節線、E 節點，最後更新 S 節點之 degree。假設此步驟總共刪去了 $\hat{k}$ 個 S 節點，而剩下 $\tilde{n}=n-\hat{k}$ 個 S 節點；計算辨識 m 條 Haplotype 所需的最少 SNP 個數 $k=\lceil \log_2 m \rceil$。

Step 3：從剩餘的 $\tilde{n}$ 個 S 節點中再挑選 $\tilde{k}=k-\hat{k}$ 個 S 節點（共有 $C_{\tilde{k}}^{\tilde{n}}$ 種可能的 S 節點候選組合），檢驗每種 S 節點候選組合中的 S 節點 degree 總合，僅保留那些滿足其 S 節點 degree 總合大於或等於剩餘的 E 節點總個數的 S 節點候選組合。

Step 4：針對 Step 3 所保留的各個 S 節點候選組合，檢驗其中的 S 節點群是否可與剩下的 E 節點完全連結；若所有的 S 節點候選組合皆無法通過此檢驗的話，表

示目前設定之 tagSNP 數尚太少，以致無法辨識所有的 Haplotype，因此我們必須增加目前的 tagSNP 個數，亦即 $\tilde{k} = \tilde{k} + 1$，並重複 Step 3；反之，只要有 S 節點候選組合可以通過此檢驗，則這些通過檢驗的 S 節點候選組合即為多重最佳解群組。

Step 5：針對每組多重最佳解群組中的每個 SNP，若其在 Step 1 中有重複或互補的其它 SNP，則可用這些重複或互補的 SNP 置換此 SNP 位置而產生另一組多重最佳解群組（值得注意的是，以此方法所產生的群組與將與其原群組之矩陣數值完全相同，但其對應的 SNP 號碼不同）。以此類推，可推導出所有的多重最佳解。

Step 6：針對每組多重最佳解，計算其 LD 估算值（r^2 或 D'），選取 LD 平均估算值最高者作為本演算法所建議之最佳解。

3.3 Multi-TagSNP 之時間複雜度

Step 1 主要執行 preprocessing 的工作，由於重複或互補的 SNP 所可以提供之辨識資訊是相同的，因此可將這些 SNP 先僅保留一個來分析，待 Step 4 算出多重最佳解群組之後，再將這些於 Step 1 被刪除的 SNP 代入置換其解中所對應的 SNP，如此即可將所有的多重最佳解全部推導出來。由於比對兩行 SNP 需耗費 $O(m)$ 時間，而全部共有 $C_2^n = O(n^2)$ 組兩兩比對，因此 Step 1 共需 $O(mn^2)$ 時間。

Step 2 先依據 3.1 小節的分析以建構 bipartite 網路圖 $G = (N, A)$，由於 degree 為 1 的 E 節點僅能被其所對應的單一之特定 S 節點來連結，因此該 S 節點勢必被選取至 tagSNP 之中，而我們則可先選取之以縮小 tagSNP 的選取範圍，加快後面的搜尋速度。由於 $|N| = |E \cup S| = O(m^2 + n)$，而 $|A| = O(m^2 n)$，因此可在 $O(\max\{|N|, |A|\}) = O(m^2 n)$ 時間內完成 G 的建構。接著演算法必須花費 $|E| = O(m^2)$ 時間以搜尋並刪除 degree 為 1

之 E 節點，並刪去與該 E 節點連結之 S 節點以及其相連之 E 節點（假設此類被刪除的 S 節點共有 $\hat{k}$ 個，則被一併刪除的 E 節點個數將介於 1 至 $\hat{k}$ 之間，此步驟將花費 $O(\hat{k})$ 時間）。由於每個 SNP 位置的數值可能為 0 或 1，因此若 k 個 SNP 最多有 2^k 種不同的 0,1 排列組合；易言之，只要是 Haplotype 個數介於 2^{k-1} 與 2^k 之間者，至少需要 k 個不同的 SNP 才能將這些 Haplotype 辨識出來。因此，辨識 m 條 Haplotype 所需的最少 SNP 個數 $k = \lceil \log_2 m \rceil$。

Step 3 與 Step 4 構成一個迴圈，此迴圈最多將執行 $O(\tilde{n} - k + \hat{k})$ 次（亦即，自 $\tilde{k} = k - \hat{k}$ 至 $\tilde{k} = \tilde{n}$）。在每次迴圈中，首先必須產生 $C_{\tilde{k}}^{\tilde{n}}$ 種可能的 S 節點候選組合，此步驟將耗費 $O(\tilde{k}C_{\tilde{k}}^{\tilde{n}})$ 的時間與空間；接著一一檢驗所產生的每個 S 節點候選組合，倘若其 S 節點 degree 總合小於剩餘的 E 節點總個數的話，則該 S 節點候選組合很明顯地將無法被用來辨識剩餘的 E 節點，因此可刪除之，此步驟亦將耗費 $O(\tilde{k}C_{\tilde{k}}^{\tilde{n}})$ 時間（因為計算每組 S 節點候選組合的 S 節點 degree 總和需要 $O(\tilde{k})$ 時間，而總共有 $O(C_{\tilde{k}}^{\tilde{n}})$ 組 S 節點候選組合）。

Step 4 必須針對 $O(C_{\tilde{k}}^{\tilde{n}})$ 組 S 節點候選組合一一檢驗其中的 S 節點群是否可與剩下的 E 節點完全連結，由於每組 S 節點候選組合內含 $\tilde{k}$ 個 S 節點與最多 $O(\tilde{n}^2)$ 個 E 節點，因此每次檢驗最多需耗費 $O(\tilde{n}^2\tilde{k})$ 時間。整體而言，Step 3 與 Step 4 可能耗費 $O((\tilde{n} - k + \hat{k})((k - \hat{k})(\tilde{n}^2 + C_k^{\tilde{n}}))$ 時間。

由於 Step 3 與 Step 4 是從 tagSNP 數量的下界開始逐一檢驗合格的 tagSNP，因此一旦發現通過檢驗的 tagSNP 候選組，演算法即可不用再提高 tagSNP 數量的下界，此時所有可通過檢驗的 tagSNP 候選組皆為本問題的多重最佳解，因此可之為多重最佳解群組。

Step 5 將 Step 1 中被刪除掉的那些數值相同或互補、而號碼卻不同的 SNP 重新代入其在每一 tagSNP 所對應的 SNP，成為另一 tagSNP。此步驟之複雜度取決於 Step 1 中所刪除的每組數值相同的 SNP 群之數量大小以及 tagSNP 的數量大小。舉例來說，

假設 Step 4 得到 Q 群多重最佳解群組，而其第 q 組 tagSNP 含有 $\breve{k}$ 個 SNP $\{S_{j_1},...,S_{j_{\breve{k}}}\}$，而各個 S_{j_u} 又與其它 w_u 個號碼不同的 SNP 具有相同或互補的數值，則全部將有 $Z(q)=\prod_{u=1,...,\breve{k}}(w_u+1)$ 組多重最佳解與該組 tagSNP 具有相同的數值；以此類推，全部將有 $\sum_{q=1,...,Q}Z(q)$ 組多重最佳解。

針對 Step 4 所求得的每一組多重最佳解群組，在 Step 6 中我們可依第二章中 LD 的估算式子（2.1）及（2.2）來估算該組 tagSNP 集合中任一元素與未被選取的 SNP 彼此間的關聯值 LD （r^2 或 D'），並對所有求出的 LD 值取平均值做為該組 tagSNP 估算的關聯程度值（涵蓋剩餘資訊量）。值得一提的是，r^2 與 D' 的計算方式僅與 SNP 的主對偶基因與次對偶基因之出現頻率數值有關，因此於 Step 5 展開同一組 tagSNP 所得到的那些多重最佳解將會具有同樣的 r^2 與 D' 數值（因所置換的 SNP 為完全相同或互補之欄位，故獲得之主、次對偶基因之出現頻率相同）。在所有的多重最佳解群組中，選取其平均 LD 值之和最大者，則該最佳解及其於 Step 5 所展開的等價最佳解群即為本演算法所推薦的最佳解，因為這些 tagSNP 可包涵更多 SNP 資訊，代表其更接近原完整 Haplotype 所涵蓋的資訊。

3.4 Multi-TagSNP 範例說明

為了更清楚地顯示 Multi-TagSNP 之演算法流程，以下舉一則小範例說明 Multi-TagSNP 演算法的每個處理過程及參數改變：首先讀取一 Haplotype 資料矩陣，以下以圖 3.5（a）中的資料矩陣 H 為例，該矩陣的 S_2, S_4, S_6 三個 SNP 以及 S_3, S_7 兩個 SNP 各具有互補的數值，根據 Step 1 中刪除與歸類之準則，將 S_4 、S_6 刪除並將之歸類至 S_2 的 group，另外還需刪除 S_7 並將之歸類至 S_3 的 group；圖 3.5（b）顯示圖 3.5（a）簡化之後的矩陣 H' 以及相關的等價 SNP 群組。

$$H=\begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{array}\begin{array}{c} \begin{array}{cccccccc} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \Rightarrow H'=\begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{array}\begin{array}{c} \begin{array}{ccccc} S_1 & S_2 & S_3 & S_5 & S_8 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$S_2\ group=\{S_4,S_6\}$

$S_3\ group=\{S_7\}$

（a）原始haplotype矩陣　　（b）刪減重複與互補SNP後之新haplotype矩陣

圖 3.5：Multi-TagSNP 演算法 Step 1 範例

Step 2 根據 2.1 小節所描述的連結規則來建構H'所對應的 bipartite 網路圖，並將所有的 S 節點與 E 節點記錄到兩個不同的集合中，此時尚未挑選出任何 tagSNP，因此 tagSNP 為空集合，其結果如圖 3.6 所示：

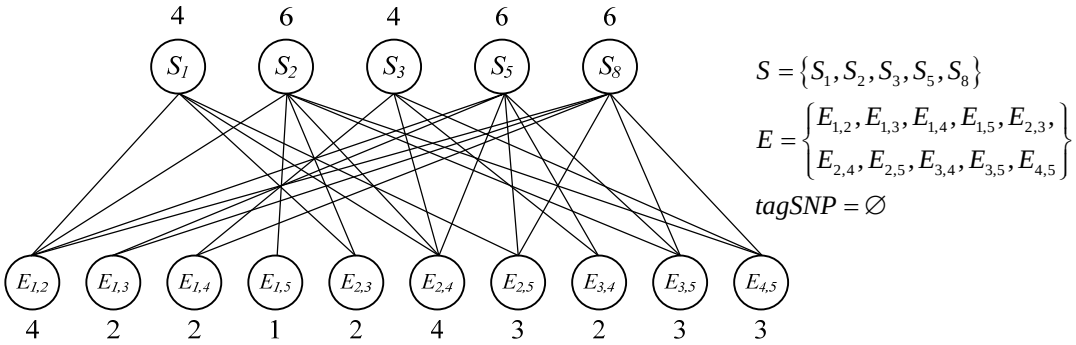


圖 3.6：Haplotype pair 比較差異對應關係圖

Step 2 將搜尋所有 degree 為 1 之 E 節點$E_{1,5}$，把$E_{1,5}$所連結的節點S_2放入 tagSNP 集合中，並將與S_2相連之節點$E_{1,2}$、$E_{1,5}$、$E_{2,3}$、$E_{2,4}$、$E_{3,5}$、$E_{4,5}$及其所有相關的節線刪除，最後將更新所有 S 節點之 degree 如圖 3.7 所示。

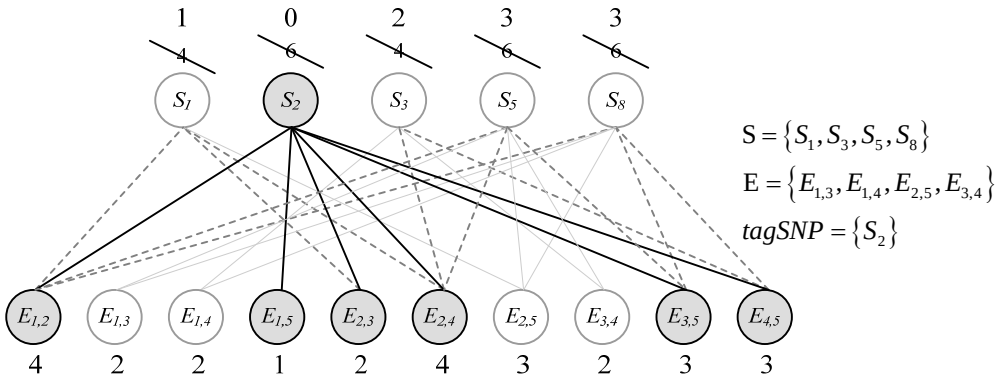
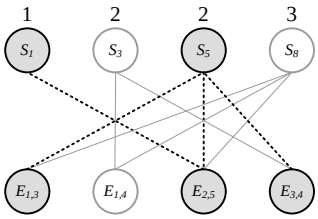


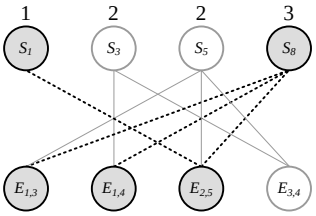
圖 3.7：Multi-TagSNP 演算法 Step 2 之處理示範圖

在 Step 3 中首先計算 $k = \lceil \log_2 5 \rceil = 3$，而此時 tagSNP 集合中有一個元素（$\hat{k} = 1$），因此再來最少僅需再從 $S = \{S_1, S_3, S_5, S_8\}$ 中挑選 $\tilde{k} = 3\text{-}1 = 2$ 個 tagSNP。可能組合為共 $C_2^4 = 6$ 組，分別為 $\{S_1, S_3\}$、$\{S_1, S_5\}$、$\{S_1, S_8\}$、$\{S_3, S_5\}$、$\{S_3, S_8\}$、$\{S_5, S_8\}$，其各組之 S 節點的 degree 總和分別為 3、4、4、5、5、6；由於目前尚有 4 個 E 節點，因此 degree 總和小於 4 之 S 節點組合一定不夠連結（亦即，辨識）所有的 4 個 E 節點，故 degree 小於 4 之組別 $\{S_1, S_3\}$ 將予以刪除。接著在 Step 4 檢驗剩餘的 5 個組別 $\{S_1, S_5\}$、$\{S_1, S_8\}$、$\{S_3, S_5\}$、$\{S_3, S_8\}$、$\{S_5, S_8\}$，看其是否可以連結剩下的 4 個 E 節點 $\{E_{1,3}, E_{1,4}, E_{2,5}, E_{3,4}\}$ 者，分別以圖 3.8（a）~（e）描繪其連結狀態。從圖中觀察得知 $\{S_3, S_5\}$、$\{S_3, S_8\}$、$\{S_5, S_8\}$ 皆可通過檢驗，分別將這三組自 S 集合中刪除並放入 tagSNP 集合中，獲得三組 tagSNP 多重解群。

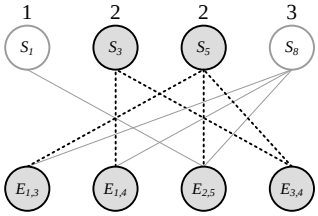
由 Step 4 可得三組多重最佳解群組：$\{S_2, S_3, S_5\}$、$\{S_2, S_3, S_8\}$、$\{S_2, S_5, S_8\}$，在 Step 5 中將 S_2 之等價 SNP 群 $\{S_4, S_6\}$ 與 S_3 之等價 SNP 群 S_7 各別代入多重解群組中的 S_2 與 S_3。舉例來說，將 S_4 置換 $\{S_2, S_3, S_8\}$ 中的 S_2 可得另一組不同的 tagSNP 解 $\{S_4, S_3, S_8\}$，以此類推總共可獲得 $3\times2\times1 + 3\times2\times1 + 3\times1\times1 = 15$ 組最佳解。



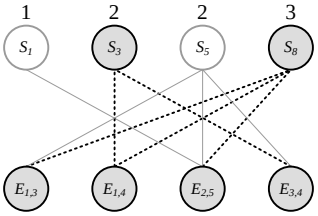
（a）{S_1,S_5}無法連結$E_{1,4}$



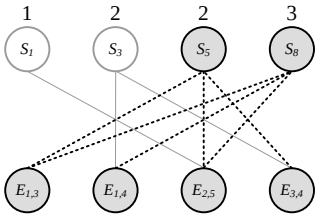
（b）{S_1,S_8}無法連結$E_{3,4}$



（c）{S_3,S_5}可連結所有的E節點



（d）{S_3,S_8}可連結所有的E節點



（e）{S_5,S_8}可連結所有的E節點

圖 3.8：Multi-TagSNP 演算法 Step 4 之處理示範圖

最後計算這 15 組最佳解之 LD 值總和，由於r^2與D'僅與 SNP 之數值有關，因此我們僅需要針對 Step 4 所得到的三組多重最佳解群組來估算其各別的r^2與D'，其平均值顯示於表 3.1，LD 估算值較高的 tagSNP 代表它比其它組 tagSNP 傳達更多的 Haplotype 序列資訊。由於 Lin（2005） 指出當 sample size 小時，D'值的誤差會很大，因此在此範例上僅列出D'值當參考，而不以D'值作為判定準則。以此例而言，$\{S_2, S_3, S_5\}$之平均r^2值為最高，代表此組解較其他兩組解為佳。

表 3.1：各組 tagSNP 之平均 r^2 值與 D' 值

tagSNP 最佳解群組	mean r^2	mean D'
$\{S_2, S_3, S_5\}$	0.2419	0.8611
$\{S_2, S_3, S_8\}$	0.2072	0.9611
$\{S_2, S_5, S_8\}$	0.2361	1

3.5 以多目標規劃模式求解 tagSNP 選取問題

3.5.1 多目標規劃模式簡介

廣義的多目標決策問題可分為兩種不同類型類，第一種為多目標決策（Multiple Objective Programming，MOP），這類問題主要在決策方案未定時，透過數學規劃模式，針對多個不同目標函數，從已知的資源限制中結合決策者的偏好資訊找到擁有最好目標值者，作為最後的決策方案。另一種類型是多屬性決策（Multiple Attribute Decision Making，MADM），又稱為多評準評估（Multi-Criteria Evaluation, MCE），此種決策問題為考量有限個不同方案之情況下，依不同的屬性特徵，對各方案做優劣排序，選擇對策者而言最理想的方案。

由 3.2 節所提出之演算法求解過程中可發現該演算法共考慮了兩種不同的求解目標，因此可被視為一種二階段的多目標最佳化問題之求解方法。亦即，該演算法先滿足極小化挑選的 tagSNP 個數之目標後，在從所求結果中選出關聯程度 LD 值之和最大者。若直接從多目標決策的角度著手，因 tagSNP 選取問題無法先獲得已知的決策方案（選取結果），較適合第一類決策問題（MOP）。因此，以下將先建立 tagSNP 選取問題之雙目標數學規劃模式，從多種常用的 MOP 解法中擇一，以演練 3.4 節中的範例，最後再驗證其求解結果與 3.4 節範例結果之差異。

3.5.2 多目標規劃 tagSNP 選取問題之數學模式

將此問題視為一個多目標規劃問題，分別考慮兩個目標函數求解 tagSNP 選取問題：第一目標將極小化所挑選的 tagSNP 個數，而另一目標則極大化已被選取的 tagSNP 與其它未被選取的 SNP 間關聯程度 LD 值之和，建立此雙目標規劃數學模式 P^{M}_{tagSNP} 如下：

$$
\begin{aligned}
Min\quad & Z(x)=\left[Z_1(x),Z_2(x)\right] && (P^{M}_{tagSNP})\\
& Z_1(x)=\sum\nolimits_{S_j\in S} x_j \\
& Z_2(x)=-\sum\nolimits_{\forall j_2>j_1} d_{j_1,j_2}y_{j_1,j_2} \\
st.\quad & \sum\nolimits_{(S_j,E_{i_1,i_2})\in A} x_j \ge 1,\ \forall E_{i_1,i_2}\in E \\
& x_{j_1}+x_{j_2}-y_{j_1,j_2}=2R_{j_1,j_2},\ \forall j_2>j_1, 1\le j_1\le n \\
& R_{j_1,j_2}\le x_{j_1} \\
& R_{j_1,j_2}\le x_{j_2} \\
& x_j\in\{0,1\},\ \forall S_j\in S \\
& y_{j_1,j_2}\in\{0,1\},\ \forall j_2>j_1, 1\le j_1\le n \\
& R_{j_1,j_2}\in\{0,1\},\ \forall j_2>j_1, 1\le j_1\le n
\end{aligned}
$$

其中 $Z_1(x)$ 為第一個求解目標，也就是極小化所挑選的 tagSNP 個數； $Z_2(x)$ 為第二個求解目標，為達到極大化關聯程度 LD 值之和的目的。為了方便起見，在此我們將第二目標轉為極小化關聯程度 LD 值之和的負數。其中 d_{j_1,j_2} 表示 S_{j_1} 與 S_{j_2} 間的關連程度 LD 值，y_{j_1,j_2} 為一個二元決策變數，當 $y_{j_1,j_2}=1$ 表示 S_{j_1} 與 S_{j_2} 中恰有一個被選取為 tagSNP；而 $y_{j_1,j_2}=0$ 則表示 S_{j_1} 與 S_{j_2} 皆被選取或皆未被選取。為達到控制 y_{j_1,j_2} 之目的，加入一中間變數 R_{j_1,j_2} 以連結 x_j 與 y_{j_1,j_2} 之關係，其中 $R_{j_1,j_2}=\min\left\{x_{j_1},x_{j_2}\right\}$。第一個限制式與模式 P_{tagSNP} 相同意義，也就是對每個 E_{i_1,i_2} 而言，至少存在某個 S_j 可使 $I(E_{i_1,i_2},S_j)=1$。第二、三、四條限制式旨在建立決策變數 x_j 與 y_{j_1,j_2} 間之關係，亦即當 $\left(x_{j_1},x_{j_2}\right)=(1,1)$ or $(0,0)$ 時，令 $y_{j_1,j_2}=1$；反之，當 $\left(x_{j_1},x_{j_2}\right)=(0,1)$ or $(1,0)$ 時，則令

$y_{j_1,j_2}=0$ 。

3.5.3 多目標規劃模式之求解方法

在求解多目標規劃問題上引入經濟學中效率的概念，根據Koopman（1951）年所提出的生產效率定義，也就是在有限資源與技術之限制下，某一財貨之產出的增加，需藉由另一財貨的減少獲得所需的資源，如此可獲得生產效率的界線。以圖3.9為例，考慮有限資源下兩種財貨X_1、X_2所有可能的生產組合可形成一塊封閉的區域，圖中虛線部分稱之為生產效率界線（productive efficiency frontier），表示資源能生產的最多X_1與X_2之組合，如圖3.9中A點與B點分別為相同生產力下不同X_1與X_2之組合，由於是有限資源所能產生的最大產生產力，因此生產效率界線上各點皆為效率（efficient）點；而生產效率界線內任一點（如點C）其生產力皆小於界線上的各點（如點A、B），因此將這些點稱之為無效率（inefficient）點。

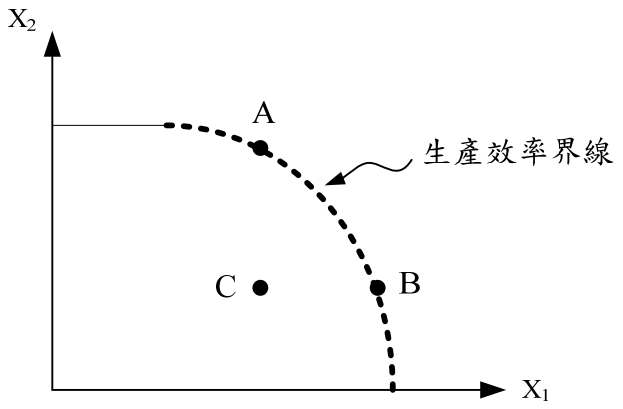


圖3.9：生產效率界線說明圖

將此觀念用於求解多目標規劃問題中，若多目標規劃模式如下：

$$\max \ \mathbf{Z} = \mathbf{CX}$$
$$st.\ \mathbf{X} \in \mathbf{S} \begin{cases} \mathbf{AX} \le \mathbf{B} \\ \mathbf{X} \ge 0 \end{cases}$$

其中**Z**表示目標空間，**C**表示準則矩陣，**X**表示決策變數，**S**表示決策空間。定義對存在的某一點$\overline{\mathbf{X}} \in \mathbf{S}$而言，若不存在另一點$\mathbf{X} \in \mathbf{S}$，使得所有的$\mathbf{CX} \ge \mathbf{C}\overline{\mathbf{X}}$，則$\overline{\mathbf{X}}$為一效率解。將決策空間轉至目標空間，決策空間中任一效率解可對應至目標空間中，將此對應點稱為非劣解（noninferior solution），其意義為對一可行解而言，若不存在它組可行解可使某一目標值增加而不需降低至少一個其他目標值，則可稱為非劣解，亦有學者稱為非超越解（nondominated solution）或是柏拉圖解（Pareto solution）。圖3.10為一個包含兩個決策變數之雙目標規劃範例圖，由圖中可看出決策空間上的效率界線（B-C-D）可完全對應至目標空間中的非劣解界線（B-C-D）。

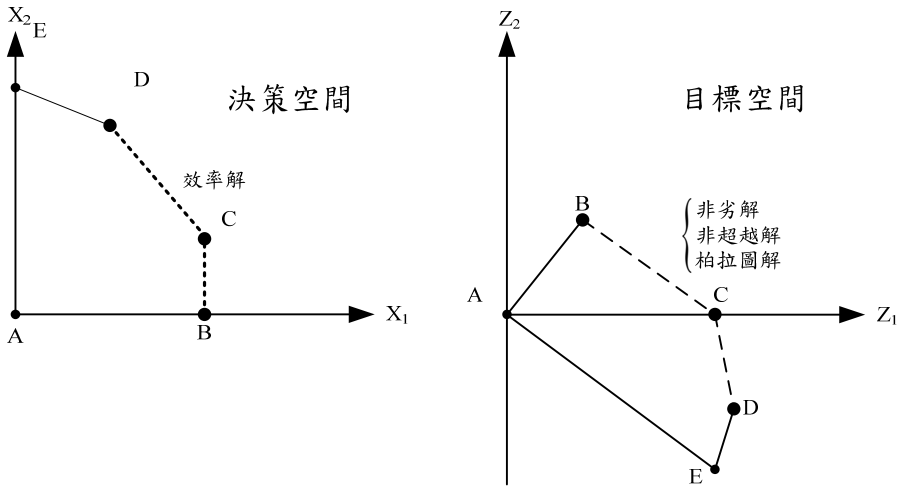


圖3.10：目標空間與決策空間之關係

一般求解多目標規劃問題可用資訊的流向分成三種不同種類的求解方式（如圖3.11所示），由左至右依序為「有偏好多目標規劃法」、「互動目標規劃法」，以及「無偏好多目標規劃法」。這三種不同的求解方式主要之差別在於偏好資訊的取得時間點，由於研究目的之一在於驗證3.2小節中求解多重解演算法的結果，因此暫先不考

慮偏好資訊，故以下僅介紹幾種無偏好多目標規劃法，每種方法各有其不同的限制。

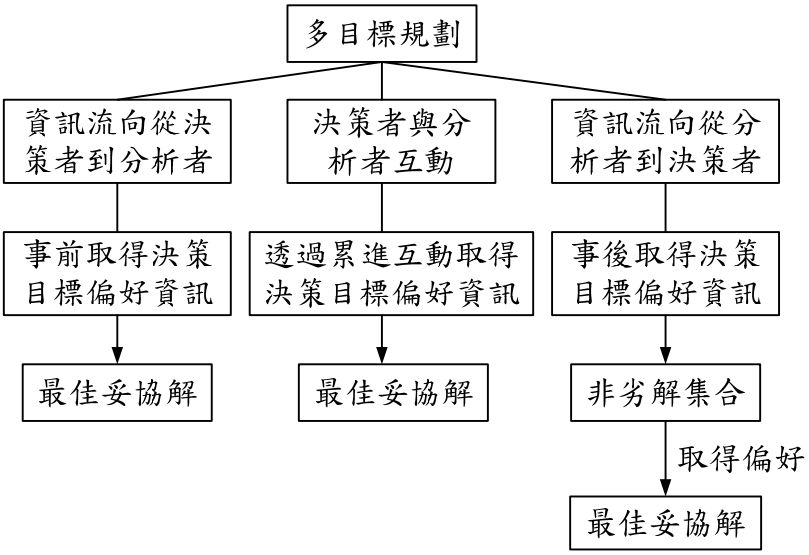


圖 3.11：依偏好資訊流向分類求解多目標規劃之方法

- **權重法（Weighting Method）**

  此方法最早由Zadeh（1963）提出，將每一目標式給予個別權重值，並假設目標可透過權數相加，在給定不同權重值下，將多個目標式組合成單一目標式，接著再求解其結果。此方法需注意其求出之結果只是近似的非劣解集合，非真正的非劣解集合，變動的權重次數越多，越逼近真正的非劣解界線，但無法得知其逼近程度為何。另外需注意權重法僅適用於可行解區域為凸集合之情況。

- **ε-限制式法（ε-Constraint Method）**

  限制其他目標值於某定值，最適化某單一目標值，藉由限制其他目標於不同定值上時，獲得一系列該單一目標值之最佳解，這些解即組合成近似之非劣解界線。有別於權重法可能發生求解跳躍之情形，此方法可用於求解可行解區域非凸集合之問題，且允許目標值的單位不同，其缺點為可能遇到在某一些限制下會造成無解之情況。

- **非劣解估計法（The Noninferior Set Estimation Method）**

  由 Cohoh（1978）提出，簡稱 NISE 法，為一種逼近非劣解集合之方法，主要藉由已知的極點之線性組合獲得新的理想解，計算其誤差是否在已定的程度內，如果違反則用新的理想解與原極點繼續推另一個理想解，如此反覆求解直到誤差在可接受程度內。此方法最大的優點在於可控制真正非劣解集合與逼近之非劣解集合間的誤差；其缺點為限制較多，僅可用於可行解區域為凸集合之問題，且只適用二維目標空間之決策問題。

3.5.4 多目標規劃 tagSNP 選取之數學模式之範例演練

採用無偏好多目標規劃之方法求解 3.4 小節之範例，其中第二目標 $Z_2(x)$ 中的 d_{j_1,j_2} 採用關係係數 r^2 值，其原因與 3.4 小節中提到的因素相同，由於此範例的規模小，其 D' 值的誤差值很大，故在此不予以採用。

為驗證 3.4 小節所求結果之正確性，以下以權重法演練相同的範例。首先，分別對兩個不同的目標函數設立不同之權數 w_1 與 w_2，接著將兩個目標函數透過權數加總組合成一個目標式，也就是將 P^{M}_{tagSNP} 的雙目標式改為：

$$Z(x) = w_1 Z_1(x) + w_2 Z_2(x) = w_1\left(\sum\nolimits_{S_j \in S} x_j\right) - w_2\left(\sum\nolimits_{\forall j_2 > j_1} d_{j_1,j_2} y_{j_1,j_2}\right)$$

代入不同的權重值，使其產生不同的權重斜率，依序求解不同權重斜率下之整數規劃問題，表 3.2 為不同的權重斜率下，其對應之效率解與非劣解，由於此範例之規模較小，依據多目標規劃中非劣解之定義，僅獲得 1 個非劣解，其求解結果選取 x_2、x_3、x_5。將此結果對照 3.4 小節的求解結果，因 3.4 小節的演算過程為先滿足第一個目標，再從中找第二目標最高者，於多目標規劃中可視為擁有較大的 w_1 值與相對較小的 w_2 值，也就是如果遇到擁有多組非劣解之問題時選擇其權重斜率值較低者；但該範例恰僅有一個非劣解，對照此組解與 3.4 小節所求之結果相同，用以驗證多目標數學模式

之正確性。

表 3.2：用權重法求解 MOP 問題範例

權重		斜率	效率解					非劣解	
w_1	w_2	w_1/w_2	X_1	X_2	X_3	X_5	X_8	Z_1	Z_2
1	0	-	0	1	1	1	0	3	1.451
5	1	-5	0	1	1	1	0	3	1.451
1	1	-1	0	1	1	1	0	3	1.451
1	5	-0.2	0	1	1	1	0	3	1.451
0	1	0	0	1	1	1	0	3	1.451

由於此範例僅有一組非劣解，不易觀察可行解滿足兩個不同目標的差異為何，因此畫出該範例所對應之目標空間圖（圖 3.12）觀察可行解滿足不同目標之程度。因該問題為一整數規劃問題，故其可行解分佈並非連續的區間而是間斷的各點。由圖 3.12 可看出當僅考慮第一個目標時（目標空間中最適化方向向左），其對應的解有 3 個，採用軟體求解時（僅算出其中一個）會因不同軟體內部運算核心的不同而求出不同的結果，因此在求解結果的驗證上，僅可比較目標值（$Z_1(x)$）的差異，不能觀察其解的差異大小；但若增加另一個目標，如 $Z_2(x)$（目標空間中最適化方向向上），則可降低多重解的發生，因同時符合兩條件的可能性較低。

由於該演練範例較小，其未選取之 SNP 資訊小於已選取的 tagSNP，因而形成 LD 值之和在圖 3.12 看起來呈現遞減的現象，這也是為何此多目標僅包含一個非劣解之原因；在一般選取 tagSNP 的問題中（未選取之 SNP 資訊遠大於已選取的 tagSNP），所遇到的情形其 LD 值之和應呈現先逐漸上升再逐漸下降之趨勢，因此所形成的非劣解集合為靠左上角的所有邊界可行解。

由圖 3.12 中亦可觀察出 3.4 小節範例的結果，由於先考慮第一目標再考慮第二

目標，搜尋圖 3.12 所有可行解中最左上的解，所找到的解亦與 3.4 小節所求結果及多目標所求解果相同。倘若遇到在其它非劣解集合元素不只一個的問題中，決策者的偏好的改變，例如允許放鬆第一目標之限制，增加第二目標之限制，最後的妥協解則可能發生改變。

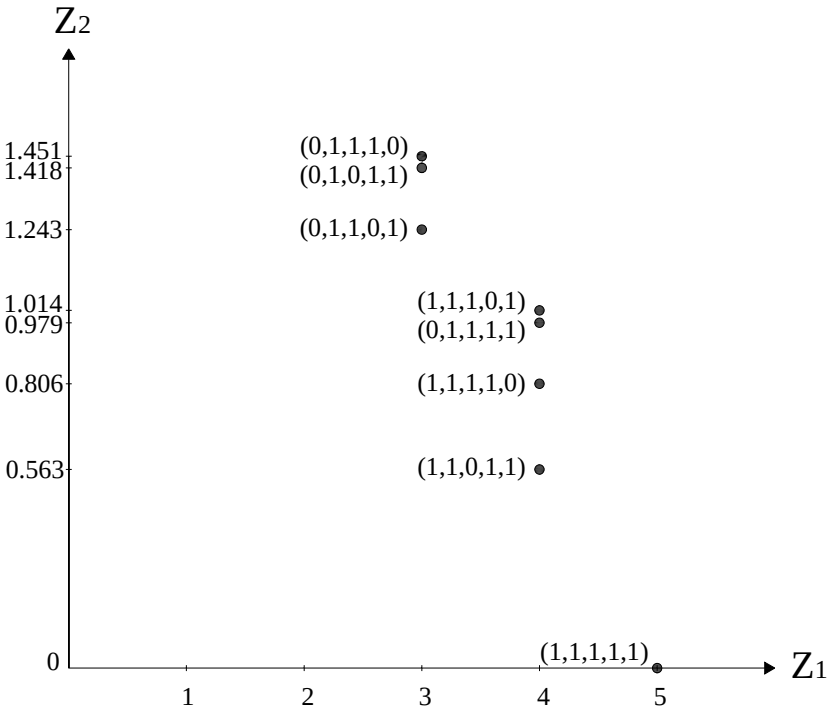


圖 3.12：MOP 演練範例之目標空間關係圖

3.6 小結

本研究提出一個求解 tagSNP 選取問題之多重最佳解方法，且為了比較多重最佳解間是否可再進一步依某些標準進行排序，我們提出一個結合連鎖不平衡觀念的求解演算法 Multi-TagSNP。由執行範例結果顯示，在這些多重最佳解中，確實存在某些具有較高 LD 值之和的 tagSNP 解，表示其所挑選的 tagSNP 與剩餘未選取的 SNP 間彼此關聯性較高，亦即該組 tagSNP 擁有較多剩餘 SNP 之資訊，因此該組 tagSNP 涵

蓋了最多原本完整 Haplotype 之序列資訊。所以我們的作法除了可用於辨識所有不同的 Haplotype 樣式之外，其所推論出整體 Haplotype 序列之可能性亦較高。

由於文獻中尚無其他學者探討多重最佳解間之差異，因此本篇研究尚無法與其它相關研究比較。 此外，由於 Multi-TagSNP 的求解過程中主要包含了兩種不同的目標，因此我們亦提出一個雙目標選取 tagSNP 的整數規劃模式 P^{M}_{tagSNP} ，並以雙目標規劃法求解 3.4 小節中 Multi-TagSNP 演練之範例，驗證其結果與 Multi-TagSNP 所求得之結果，並闡述兩種不同目標值的意義。

由於本章節所提之 Multi-TagSNP 演算法僅適合用於較小規模的 tagSNP 選取問題，因此可將此種規模的問題視為已將原選取問題切割多個 Block 後的資料型態。在求取較大規模的 TagSNP 選取問題方面，本研究將於第四章提出一個以拉氏鬆弛法為基礎的啟發式演算法，其與第三章不同之處在於該演算法僅考慮選取最少個 tagSNP 個數以辨識 Haplotype 樣式的差異，而不會結合連鎖不平衡的概念，且其演算法最後僅求出一個近似解而非多重解。

第四章 以拉氏鬆弛法求解 tagSNP 選取問題

本章提出一個以拉氏鬆弛法為基礎之啟發式演算法 LRH 以求解 tagSNP 選取問題。4.1 小節將概述基本拉氏鬆弛法之演算流程，詳述每一步驟之重要事項；接著在 4.2 小節中描述一些加快演算速率與求解品質之方法，合併 4.1 小節與 4.2 小節之演算法命名為 LRH。除此之外，我們亦結合 LRH 與 CPLEX 兩種求解方法之優點，提出一命名為 MIX 的二階段演算法以改善求解效率品質；在 4.3 小節首先說明測試的資料樣本，接著呈現各種資料其 CPLEX、LRH 與 MIX 的比較結果。

4.1 拉氏鬆弛法

4.1.1 拉氏問題模式

將 tagSNP 選取問題原數學模式 P_{tagSNP} 轉換為放鬆後的拉氏鬆弛模式如下，其中以 $u_{E_{i_1,i_2}}$ 代表拉氏乘數（Lagrangian multiplier）。

$$L(u) = \min \sum_{S_j \in \mathrm{S}} x_j + \sum_{E_{i_1,i_2} \in E} u_{E_{i_1,i_2}} \left(1 - \sum_{(S_j, E_{i_1,i_2}) \in A} x_j\right)$$
$$st.\ \ x_j \in \{0,1\},\ \forall S_j \in S$$

定義 I_j 為單一 SNP_j 可涵蓋的 E_{i_1,i_2} 所構成的集合（亦即 $I_j = \left\{E_{i_1,i_2} \in E : (S_j, E_{i_1,i_2}) \in A\right\}$）；$J_{i_1,i_2}$ 為對單一 E_{i_1,i_2} 而言，涵蓋此 E_{i_1,i_2} 的所有 SNP_j 構成之集合（亦即 $J_{i_1,i_2} = \left\{S_j \in S : (S_j, E_{i_1,i_2}) \in A\right\}$）。由於此問題所有變數之係數僅包含 0 與 1，因此可將上述模式整理成模式 P^L_{tagSNP} 如下，其中 **u** 代表所有 $u_{E_{i_1,i_2}}$ 所形成之向量，$c_j(\mathbf{u}) = c_j - \sum_{E_{i_1,i_2} \in I_j} u_{E_{i_1,i_2}}$。

$$L(u) = \min \sum_{S_j \in S} c_j(\mathbf{u}) x_j + \sum_{E_{i_1,i_2} \in E} u_{E_{i_1,i_2}} \qquad \left(\mathrm{P}^L_{tagSNP}\right)$$
$$st.\ \ x_j \in \{0,1\},\ \forall S_j \in S$$

將 P_{tagSNP}^{L} 依拉氏鬆弛法求解，根據對偶定理，求解拉氏問題所得之目標函數值可視為原問題之下界，透過不斷提昇下界方式使其逐漸逼近原問題之最佳解。求解方式一般使用次梯度法，此方法藉由不斷修正拉氏乘數，直到找到一組拉氏乘數使 P_{tagSNP}^{L} 獲得最佳值。

4.1.2 次梯度法

從文獻回顧中可得知，拉氏鬆弛法主要藉由上下界夾擠的方式來尋找精確解。由 P_{tagSNP}^{L} 模式可得知其 $\sum_{E_{i_1,i_2}\in E} u_{E_{i_1,i_2}}$ 可藉由更新拉氏乘數而得，因此可視為一已知值，故 $L(u)$ 大小僅受 $\min \sum_{S_j\in S} c_j(\mathbf{u}) x_j$ 影響；再者，由於 x_j 是binary變數，因此可藉由控制 x_j 的值（0或1）以決定係數 $c_j(\mathbf{u})$ 的選取方式；換言之，如果某 $c_j(\mathbf{u})$ 對限制式有貢獻，則將其對應之 x_j 設為1；反之，則將 x_j 設為0：

$$\begin{cases} x_j(u)=1,\ if\ c_j(u)<0 \\ x_j(u)=0,\ if\ c_j(u)>0 \\ x_j(u)=0\ or\ 1,\ if\ c_j(u)=0 \end{cases} \tag{4.1}$$

透過式子（4.1）可很快獲得 $\min \sum_{S_j\in S} c_j(\mathbf{u}) x_j$ 之值，因此如欲藉由極大化 $L(u)$ 以提升下界，則需不斷調整**u**來獲得 $\max\ L(u)$，根據文獻回顧得知採用次梯度法可快速尋找到接近最佳乘數向量，而該方法以遞迴方式更新拉氏乘數，產生一序列非負之乘數向量 $\mathbf{u}^0,\mathbf{u}^1,\ldots,\mathbf{u}^k$ ，本研究採用貪婪法則的概念（Caprara et al.，1999）將拉氏乘數之初始值定義為

$$u_{E_{i_1,i_2}}^{0} = \min_{S_j\in J_{i_1,i_2}} \frac{1}{|I_j|},\ E_{i_1,i_2}\in E \tag{4.2}$$

而 $\mathbf{u}^k$ 則表示第 k 次拉氏乘數更新後之向量，因此 P_{tagSNP}^{L} 之拉氏乘數更新函數（Held and Kap，1971；Caprara et al.，1999）為：

$$u_{E_{i_1,i_2}}^{k+1} = \max\left\{ u_{E_{i_1,i_2}}^{k} + \lambda \frac{UB - L\left(\mathbf{u}^k\right)}{\left\| s\left(\mathbf{u}^k\right) \right\|^2} s_{E_{i_1,i_2}}\left(\mathbf{u}^k\right), 0 \right\}, \; for \; E_{i_1,i_2} \in E \tag{4.3}$$

其中 $s_{E_{i_1,i_2}}(\mathbf{u}) = 1 - \sum_{S_j \in J_{i_1,i_2}} x_j(\mathbf{u}),\ E_{i_1,i_2} \in E$ ，可視為 E_{i_1,i_2} 被 SNP_j 涵蓋之次數，其實質意義如下：

$$s_{E_{i_1,i_2}}(\mathbf{u}) \begin{cases} <0, & \text{表示被涵蓋超過1次} \\ =0, & \text{表示被涵蓋恰為1次} \\ >0, & \text{表示沒有被涵蓋到} \end{cases} \tag{4.4}$$

而*UB*所代表的是遞迴運算過程中出現過的最佳上界值，$L\left(\mathbf{u}^k\right)$ 是第 k 次運算所求之下界值，其所對應的解為放鬆問題的現行解。*UB*與 $L\left(\mathbf{u}^k\right)$ 所夾擠之區間為最佳解所在的區間範圍。

當上下界值接近時，也就是所夾擠之區間 $UB - L\left(\mathbf{u}^k\right)$ 較小時，縮減調整的幅度可較快獲得收斂的解。由於λ為一個給定的幅度參數（step length），$s_{E_{i_1,i_2}}(\mathbf{u})$ 是**u**值更新的方向，$UB - L\left(\mathbf{u}^k\right)$ 為更新的範圍限制，因此幅度參數λ值的設定可對*UB*、$L\left(\mathbf{u}^k\right)$ 與 $s_{E_{i_1,i_2}}(\mathbf{u})$ 三項所構成的更新程度作細部調整，避免修正幅度過大或過小造成收斂的效率很差。另外，$\left\| s\left(\mathbf{u}^k\right) \right\|^2$ 表示 $s_{E_{i_1,i_2}}(\mathbf{u})$ 的長度平方，由於該向量長度平方值越大時，表示欲前進的方向誤差越大，因此將此值放置於分母，藉以控制調整的幅度不要過大。

4.1.3 演算流程

圖4.1概述整個演算法的流程，首先讀入所欲求解的問題資料，接著設定變數之初始值。由於考慮在tagSNP選取問題中的最糟情況即為選取所有的SNP，且拉氏乘數之初始值若採用貪婪法將可在其下一步驟隨即獲得一組對應解，而此時若其檢驗式 $s_{E_{i_1,i_2}}(\mathbf{u})$ 可符合原問題限制，則可立即修正上界值，因此我們決定將初始上界值*UB*直接先設定成全部SNP之總個數，而不考慮以某種啟發式演算法求得一解再由其訂定初始上界值。

在求解拉氏問題的步驟上，根據式子（4.1）可快速獲得其對應的解，再依該組解可得到新的$s_{E_{i_1,i_2}}(\mathbf{u})$；從式子（4.4）可檢驗是否有哪組$E_{i_1,i_2}$尚未被涵蓋到，如果有任意一個$E_{i_1,i_2}$仍未被涵蓋到，則代表此組解必不符合原問題限制；如果通過檢驗，則將此組解與其對應的$L(\mathbf{u}^k)$、$c_j(\mathbf{u})$以及$u_{E_{i_1,i_2}}$記錄下來。

判定是否符合原限制式後，將通過檢驗之解的選取數目總和與原*UB*值做比較，若選取數目總和較*UB*值低，則將*UB*值更新為現行解之選取數目總和；反之則不做更動。接著依據式子（4.3）更新拉氏乘數$\mathbf{u}^k$。

從求解拉氏問題到更新拉氏乘數，不斷反覆這些步驟直到所訂定的遞迴次數上限為止，此步驟的遞迴次數上限可由使用者依據不同問題規模與求解的品質來訂定。除此之外，尚可訂定其它的終止條件，譬如訂定上下界值的差異不超過某一設定值，或者採用當連續幾次更新後改善效率不高於某設定值即停止。

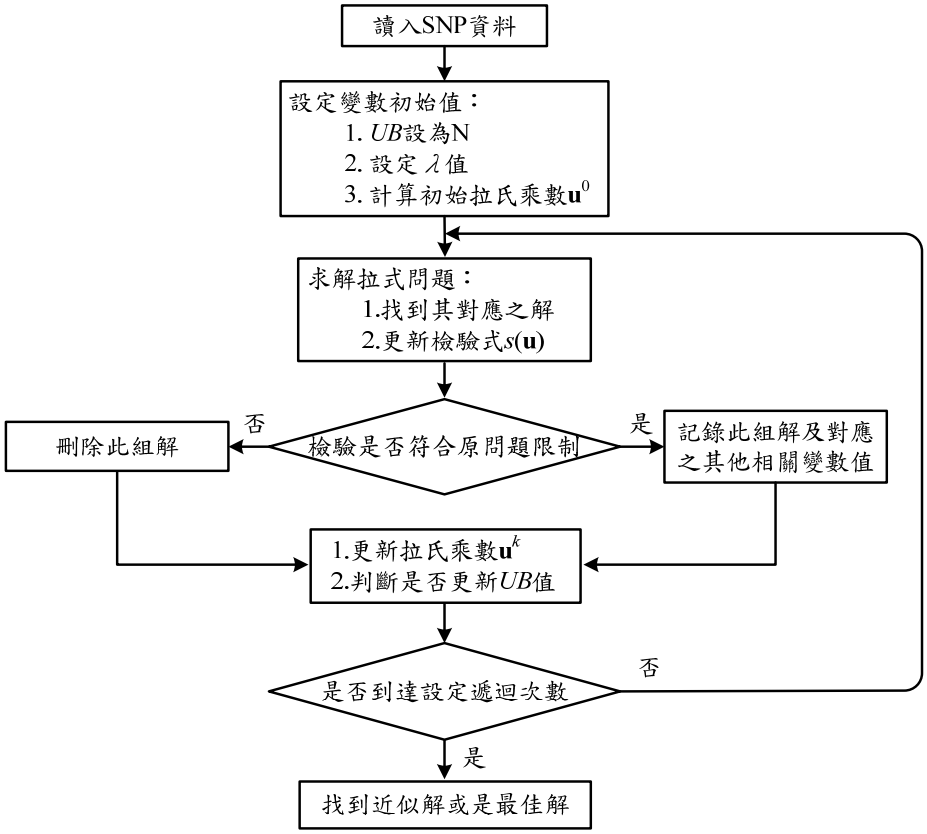


圖 4.1：拉氏鬆弛法之演算流程圖

4.1.4 範例說明

首先給定 SNP 的輸入資料，這些資料需先經過前置處理，由於此演算法旨在完全辨識所有不同的 Haplotype，因此需先刪除完全相同的列，接著執行第三章求解多重解的演算法類似步驟，先將 degree 為 1 的 E_{i_1,i_2} 搜尋出來，將該 E_{i_1,i_2} 對應的 SNP_j 先放入 tagSNP（換言之，先將這些 SNP_j 欄位刪除），相同或完全互補的欄位可在前置處理時先刪除，或者留到演算法的最後一個程序（亦即找到近似解時）再刪除；以上的前置處理完成後再進入下一個演算流程。此範例採用圖 3.2 中的矩陣 *EH*（可當成已處理過的資料）說明，讀入 SNP 資料後接著是變數的設定，由於此問題的 SNP 共有 5 個，因此設定 *UB* 的初始值為 5。在 λ 值的設定上，因為該範例為一個小規模問題，因此簡單設定 λ 值為 0.1；在大規模問題上，可訂定一些 λ 值的調整規則以改善收斂效率。接著是計算拉氏乘數向量的初始值，根據式子（4.2）可計算每一個 $u^0_{E_{i_1,i_2}}$，因此可得：

$$u^0_{E_{1,2}} = \min\left\{\frac{1}{|I_1|},\frac{1}{|I_2|},\frac{1}{|I_5|}\right\} = \min\left\{\frac{1}{\left|\{E_{1,2},E_{2,3},E_{2,4}\}\right|},\frac{1}{\left|\{E_{1,2},E_{1,3},E_{1,4}\}\right|},\frac{1}{\left|\{E_{1,2},E_{1,3},E_{2,4},E_{3,4}\}\right|}\right\}$$
$$= \min\left\{\frac{1}{3},\frac{1}{3},\frac{1}{4}\right\} = \frac{1}{4}$$

，以此類推可得 $u^0_{E_{1,3}} = \frac{1}{4}$、$u^0_{E_{1,4}} = \frac{1}{4}$、$u^0_{E_{2,3}} = \frac{1}{4}$、$u^0_{E_{2,4}} = \frac{1}{4}$、$u^0_{E_{3,4}} = \frac{1}{4}$。

接下來進入求解拉氏問題步驟，首先建立 P^L_{tagSNP} 模式如下：

$$L(u) = \frac{1}{4}x_1 + \frac{1}{4}x_2 + 0\cdot x_3 + \frac{1}{4}x_4 + 0\cdot x_5 + \left(\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{4}\right)$$

由式子（4.1）判斷其對應解為 $\mathbf{x} = (0,0,1,0,1)$，因此可得 $L(u) = 1.5$，接下來更新檢驗式 $s(\mathbf{u})$。以 $s_{1,2}(\mathbf{u})$ 為例，$s_{1,2}(\mathbf{u}) = 1-(1,1,0,0,1)\cdot(0,0,1,0,1) = 0$，以此類推可得 $s(\mathbf{u}) = \{0,\text{-}1,0,0,\text{-}1,0\}$。由 $s(\mathbf{u})$ 檢驗依照（4.4）式檢驗是否符合原限制式規定，也就是對每個 $E_{i_1,i_2} \in E$ 而言，至少會被涵蓋一次，因此檢驗式中的每個數值皆需不大於 0。

就此範例而言，對每一 $E_{i_1,i_2} \in E$ ，$s_{E_{i_1,i_2}}(\mathbf{u}) \le 0$ ，代表此組解符合原問題之限制式，因此將該組解記錄下來，且因 $x_j = 1$ 之個數為 2，小於目前的 *UB* 值，因此將 *UB* 值修改為 2。

下一個步驟進入拉氏乘數的更新，將 $\mathbf{u}^0 = \left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right)$ 、$\lambda = 0.1$ 、$UB = 5$ 、$L(u) = 1.5$ 、$s(\mathbf{u}^0) = \{0,-1,0,0,-1,0\}$ 與 $\left\|s(\mathbf{u}^0)\right\|^2 = 2$ 代入（4.3）式，得到新的拉氏乘數向量 $\mathbf{u}^1 = (0.25, 0.075, 0.25, 0.25, 0.075, 0.25)$。由於尚未達到遞迴次數上限因此進入求解拉氏問題步驟，不斷反覆此步驟開始至拉氏乘數更新的遞迴直到終止條件發生。依此範例而言若不限制遞迴次數，在 $\lambda = 0.1$ 時，執行 212 次之後其下界與上界會相同（也就是出現最佳解）；在 $\lambda = 0.5$ 時，共執行 33 次；在 $\lambda = 1.5$ 時，僅需執行 18 次；而當在 $\lambda = 2$ 時，又需執行 33 次，由此可知 λ 值的調整的確會影響收斂速度。

4.2 啟發式拉氏鬆弛演算法 LRH

為增加求解品質與求解效率，本研究對原拉氏演算法流程作部分修改。在求解拉氏問題的步驟上，當已知 *UB* 值的情況下，表示現行的所有解中至少有一個可行解僅選取 *UB* 個 tagSNP，將此解視為目前的最佳解；若欲尋求新的理想解，照理說所求之解不應超過上界值，根據此目的我們將修正此演算法流程中對應解的挑選方式，而不是僅採用式子（4.1）的判斷方式。

當符合式子（4.1）的個數超過現行的 *UB* 值時，先將所有符合式子（4.1）之 $c_j(\mathbf{u})$ 值與其對應的 x_j 依照大小排序出來，僅將與 *UB* 同個數的 x_j 值設為為 1。以下以範例說明，從圖 4.2 中的式子可看出如依照式子（4.1）將選取 x_3 、x_4 、x_6 與 x_7 四個變數設為 1，但此時 *UB* 值僅為 3，故如圖 4.2 先將每一項 $c_j(\mathbf{u})$ 值與其對應之 x_j 依序排列，僅挑選前 3 個 x_j 變數，亦即將 x_4 、x_7 與 x_6 設為 1。

$$\sum_{S_j \in S} c_j(\mathbf{u}) x_j = x_1 + 0.12x_2 + (-0.04)x_3 + (-0.08)x_4 + 0.13x_5 + (-0.07)x_6 + (-0.08)x_7 + 0.12x_8$$

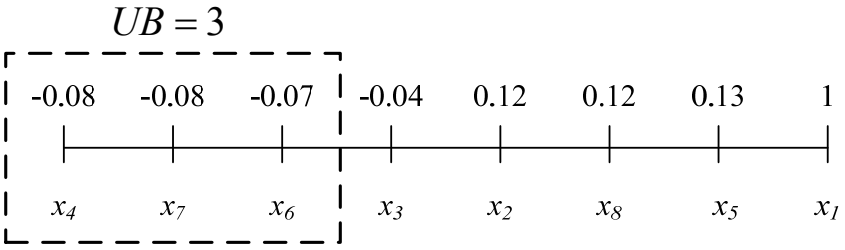


圖 4.2：修正演算法之挑選對應解範例說明

另外，我們從測試的案例發現：在求解小規模問題時，我們演算法所求得之解與 CPLEX 解出之精確解近乎相同；而當問題規模越大時，演算法所求之解與精確解差異越大。因此我們可推論問題規模的縮減可改善求解的品質。類似想法亦可見於 Caprara et al.（1999），他們在求解 SCP 之演算流程中採用固定欄位的方式縮減問題規模，因此本研究亦加入了一些固定欄位的規則以縮減問題規模並期能進一步改善求解品質。

首先，我們比較現行解與前次解的差異，將兩組解所選取的相同 x_j 記錄下來（以下以 x_j' 表示）。如圖 4.3 所示，其中 x_4、x_5 皆出現於前次解與現行解，因此將其記錄下來（x_4'、x_5'）。接下來搜尋 E_{i_1,i_2} 之 degree 最小者（如多組具有同樣最小 degree 則任選其一），比較可涵蓋該 E_{i_1,i_2} 之 SNP_j 與 x_j' 對應的 SNP_j' 是否有相同的部分；若僅有一個相同，則將該欄位固定（刪除 SNP_j' 欄位，並將該 SNP_j' 加入 tagSNP）；若有多個相同則選取 SNP_j' 之 degree 最大者，當遇到 degree 最大者亦有多組時則又任選其中一個。以下以範例說明，從圖 4.3 中矩陣 EH 可知 $E_{1,3}$ 的 degree 最小，可被 x_5 涵蓋，而記錄中 x_5' 的前次解與現行解恰好相同，因此固定 x_5'（刪除 SNP_5' 欄位，將該 SNP_5' 加入 tagSNP）；倘若有多個符合條件者（如圖 4.4 中的 x_4' 與 x_5'），則比較其 degree 大小，最後固定 degree 較大者之欄位（x_4'）。

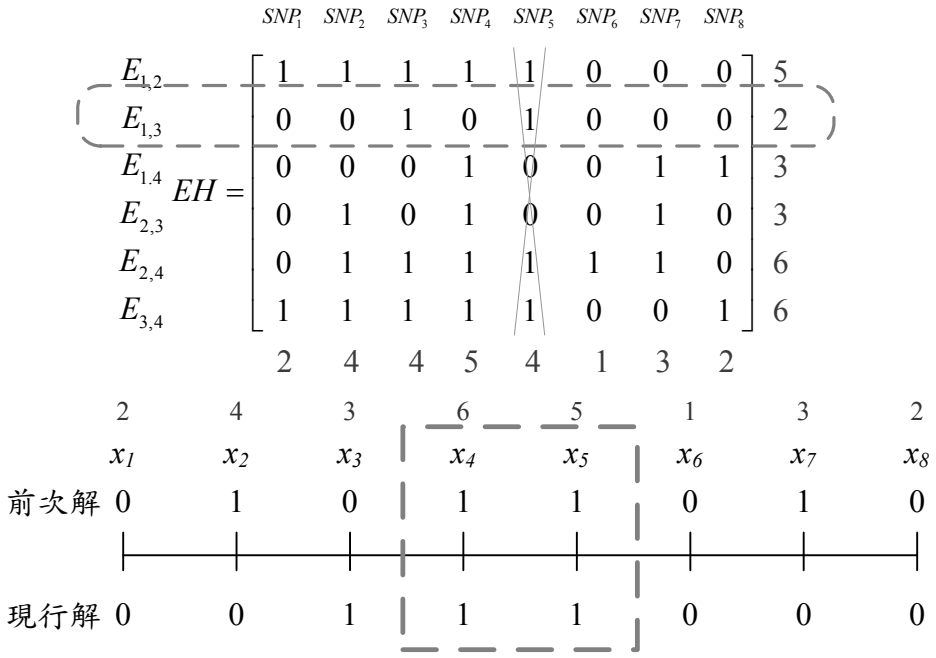


圖 4.3：Case 1 固定欄位範例說明

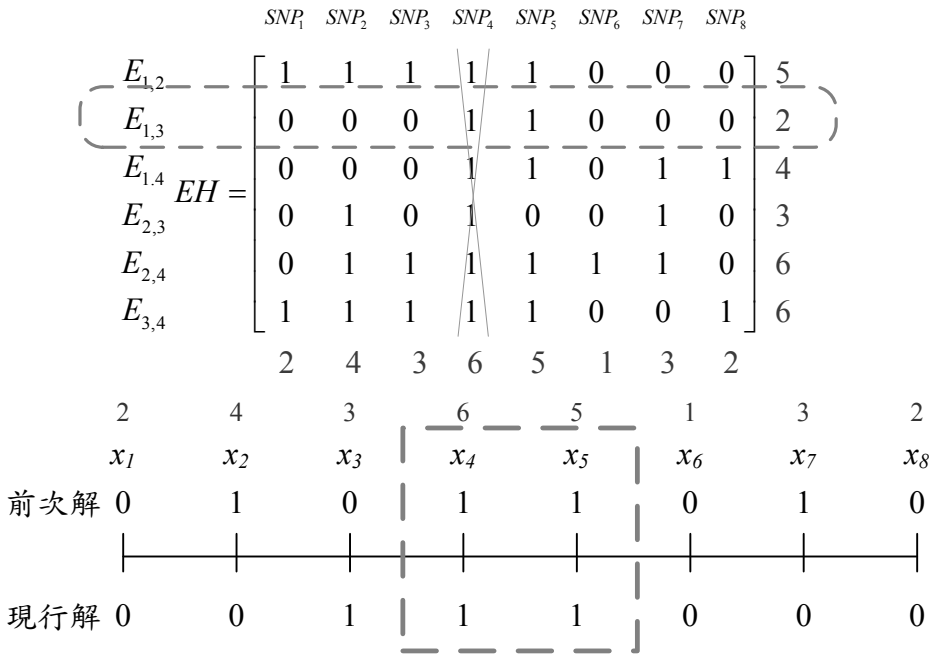


圖 4.4：Case 2 固定欄位範例說明

若無任何符合條件之 SNP_j 可選，則我們僅視被記錄的 SNP_j' 之 degree 大小來決定要固定之欄位。以圖 4.4 為例，$E_{1,3}$ 的 degree 最小者可被 SNP_3、SNP_7 涵蓋，與記錄

的SNP_4'、SNP_5'無任何相同部分，因此僅刪除SNP_4'、SNP_5'之 degree 較大者之欄位。以此例而言，最終將固定SNP_4'，並將SNP_4'加入 tagSNP。

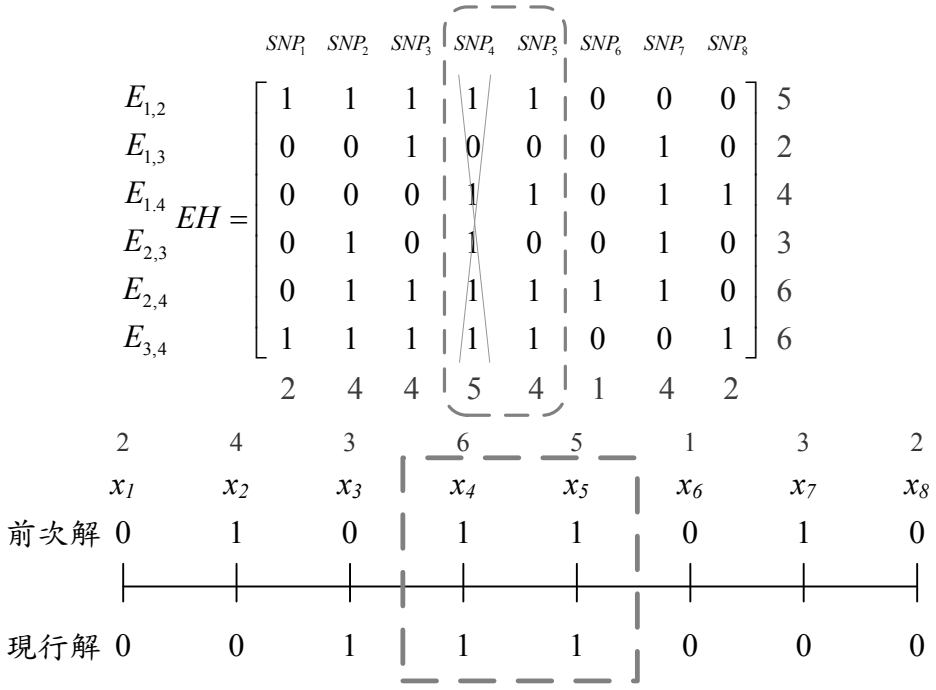


圖 4.5：Case 3 固定欄位範例說明

將 4.1 小節與 4.2 小節所描述之演算流程整合成一個啟發式拉氏鬆弛法（Lagrangian Relaxation Heuristic，LRH），將此演算法命名為 LRH。修正條件的加入不僅可改善求解的品質，亦可減少資料的前置處理。由於一次僅固定一個欄位，因此若兩個欄位有相同的辨識能力（完全相同或完全互補），僅有其中一個可能被固定（選取），且該被固定之欄位其辨識能力將被完全反應在判斷式$s_{E_{i1,i2}}(\mathbf{u})$中，在下一個循環中的更新方向會隨之改變，以降低選取完全相同或互補的欄位的可能性。

4.3 LRH 演算法數值測試分析

4.3.1 測試資料

過去學者的研究中用來測試的資料樣本一般而言都不會太大，若欲求解大規模的選取問題（如 Patil et al.（2001）求解人類第 21 號染色體，共包含 24047 個 SNP，20 組不同的 Haplotype）時，一般會先對該問題根據不同的 Block 定義做切割，再對切割後的 Block 進行選取 tagSNP，此時每一 Block 中所包含的 SNP 個數較少。舉例來說，Patil et al.（2001）將 24047 個 SNP 切割了 4135 個 Block，其 Block 中最長序列有 103 個 SNP；Avi-Itzhak（2003）則分別測試兩個不同人種各 45 個體（即 45 個 Haplotype）其 6 號染色體、21 號染色體及 22 號染色體，在給定 Block 切割情形後求解 tagSNP 選取問題，其測試的資料中最長的 Block 包含 22 個 SNP；Huang et al.（2005）除了測試 Patil et al.（2001）所求之資料外，另測試 10 個 Haplotype 且分別包含 20 個 SNP 與 40 個 SNP 的隨機產生資料與由 Hudson（2002）模擬器所產生之資料。

參考文獻中各種測試資料及其規模之後，本論文將分別針對模擬資料與真實資料兩種資料進行數值測試。

根據一般的求解經驗，除非包含的 SNP 個數不多，否則超過 20 個 Haplotype 的 tagSNP 選取問題很難在可忍受的有限時間內獲得最佳解；然而，LRH 之主要目的即在於解決大規模的 tagSNP 選取問題，因此我們將模擬資料分成九種不同規模（規模最大之問題尚於 CPLEX 可忍受之求解時限內），分別測試 15 個、30 個及 45 個 Haplotype 的 tagSNP 選取情形，而每一 Haplotype 則又分別包含 50 個、100 個及 150 個 SNP。

在產生模擬資料方面，我們採用兩種不同模擬器。第一類模擬資料參考生物理論，以基因序列中每一基因座上主對偶基因與次對偶基因的關係及其出現頻率產生模擬資料，由於過去研究統計主對偶基因出現頻率高於 15%，因此將超過 15%之模擬

亂數設定為主對偶基因；依此方式決定序列中的每一基因座之主對偶基因及次對偶基因，接著設定每一基因座之突變率並根據已決定的主、次對偶基因模擬序列資料，本論文將突變率設為 20%與 5%（以下分別以 $P_{m\times n}^{Sim0.05}$ 與 $P_{m\times n}^{Sim0.2}$ 表示該類資料）。第二類模擬資料採用 Hudson（2002）模擬器產生之資料，以下以 $P_{m\times n}^{Hudson}$ 表示該類資料。在真實資料上採用 Patil et al.（2001）測試之資料，延用其切割 Block 之資訊以求解切割 Block 後之 tagSNP 選取問題與未切割 Block 之大規模 tagSNP 選取問題。

4.3.2 測試結果

4.3.2.1 模擬資料之測試結果

以 LRH 演算法求解不同規模的選取問題，以 $P_{m\times n}^{Data}$ 表示三種不同模擬 Data （其中 Data 包含 Sim0.05、Sim0.2、Hudson）下包含 m（m=15,30,45）個 Haplotype，n（n=50,100,150）個 SNP；舉例說明，$P_{30\times 50}^{Sim0.05}$ 表示模擬包含 30 個 Haplotype，50 個 SNP（亦即 m = 30,n = 50）突變率為 5%的資料，以此類推共有 3^3=27 種資料樣本，對每種樣本各測試 10 個案例。

圖 4.6、圖 4.7 與圖 4.8 為採用 LRH（$\lambda = 0.4$）測試三類資料（$P_{m\times n}^{Sim0.05}$、$P_{m\times n}^{Sim0.2}$ 與 $P_{m\times n}^{Hudson}$）的收斂情形，為了清楚觀察不同 iteration 下，LRH 選取之 tagSNP 個數的收斂狀況，橫軸座標以對數（log）方式呈現以拉開下降的幅度。從圖 4.6、圖 4.7 與圖 4.8 皆可觀察得到 LRH 在求解的初期其收斂效率優異，後期的收斂效率較差，當遞迴次數為 1000 次時，所求的結果已近似最後的收斂解，因此後續 LRH 之測試結果皆將遞迴次數設為 1000 次。

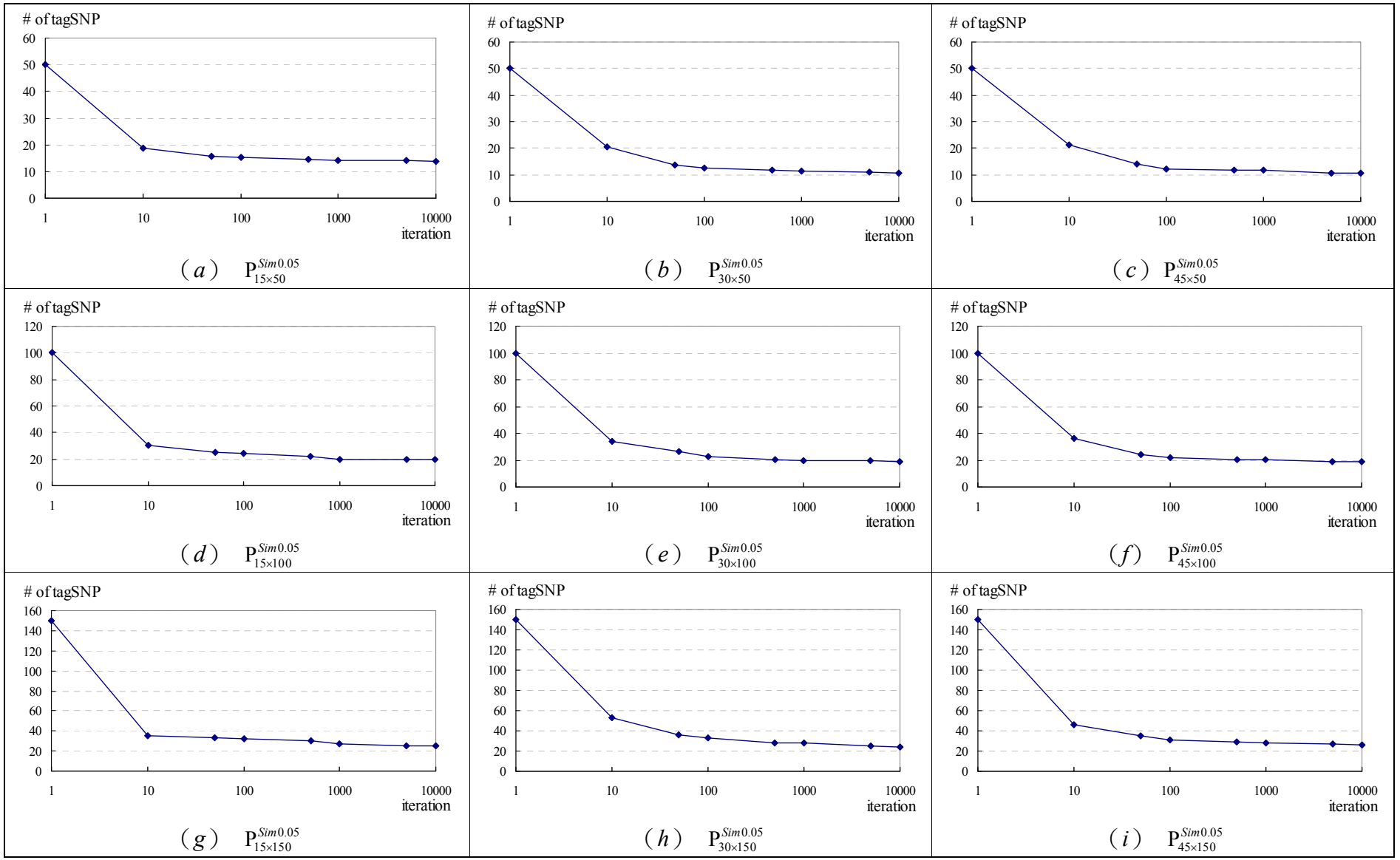


圖 4.6：採用 LRH 求解 Sim0.05 資料之收斂情況

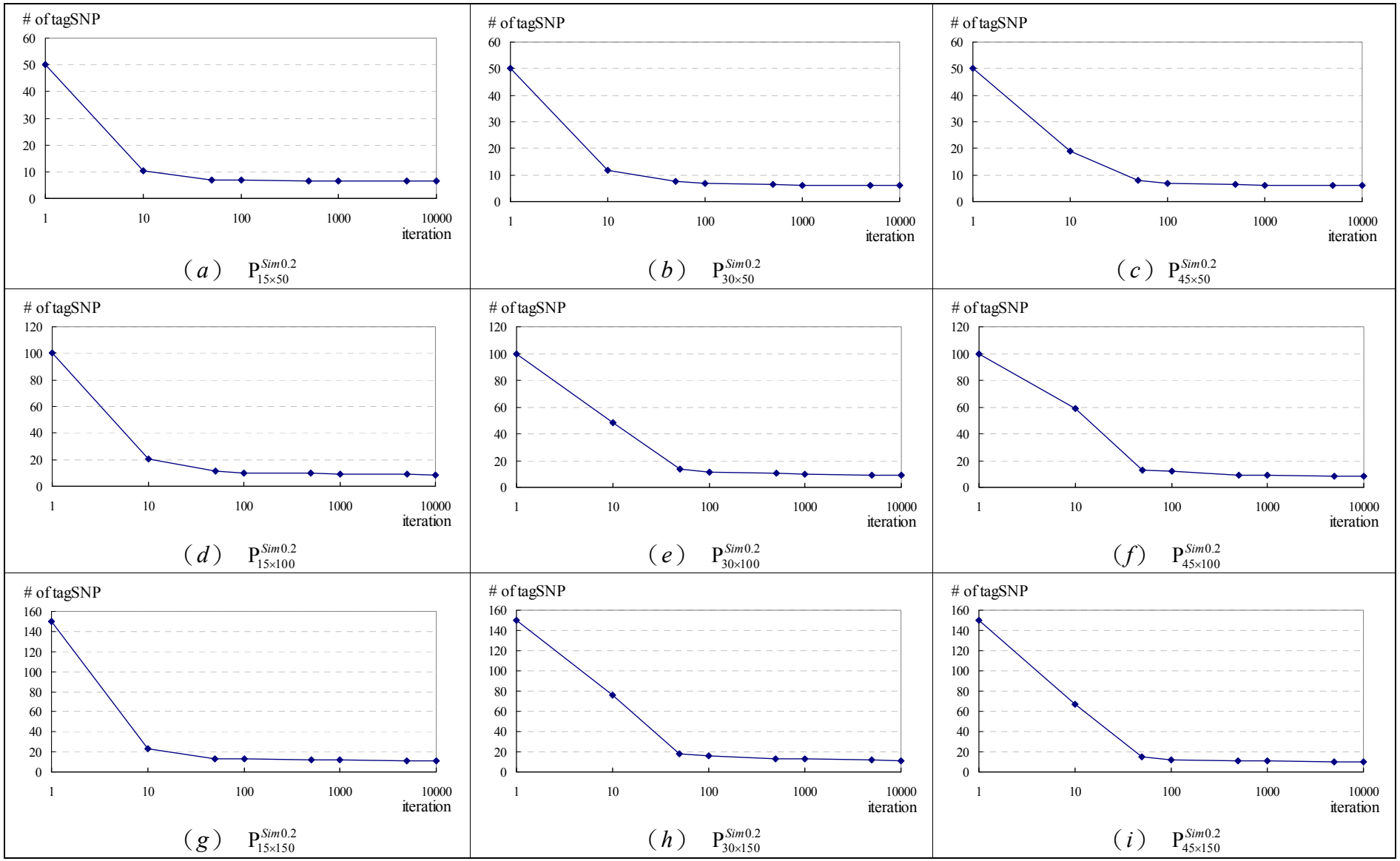


圖 4.7：採用 LRH 求解 Sim0.2 資料之收斂情況

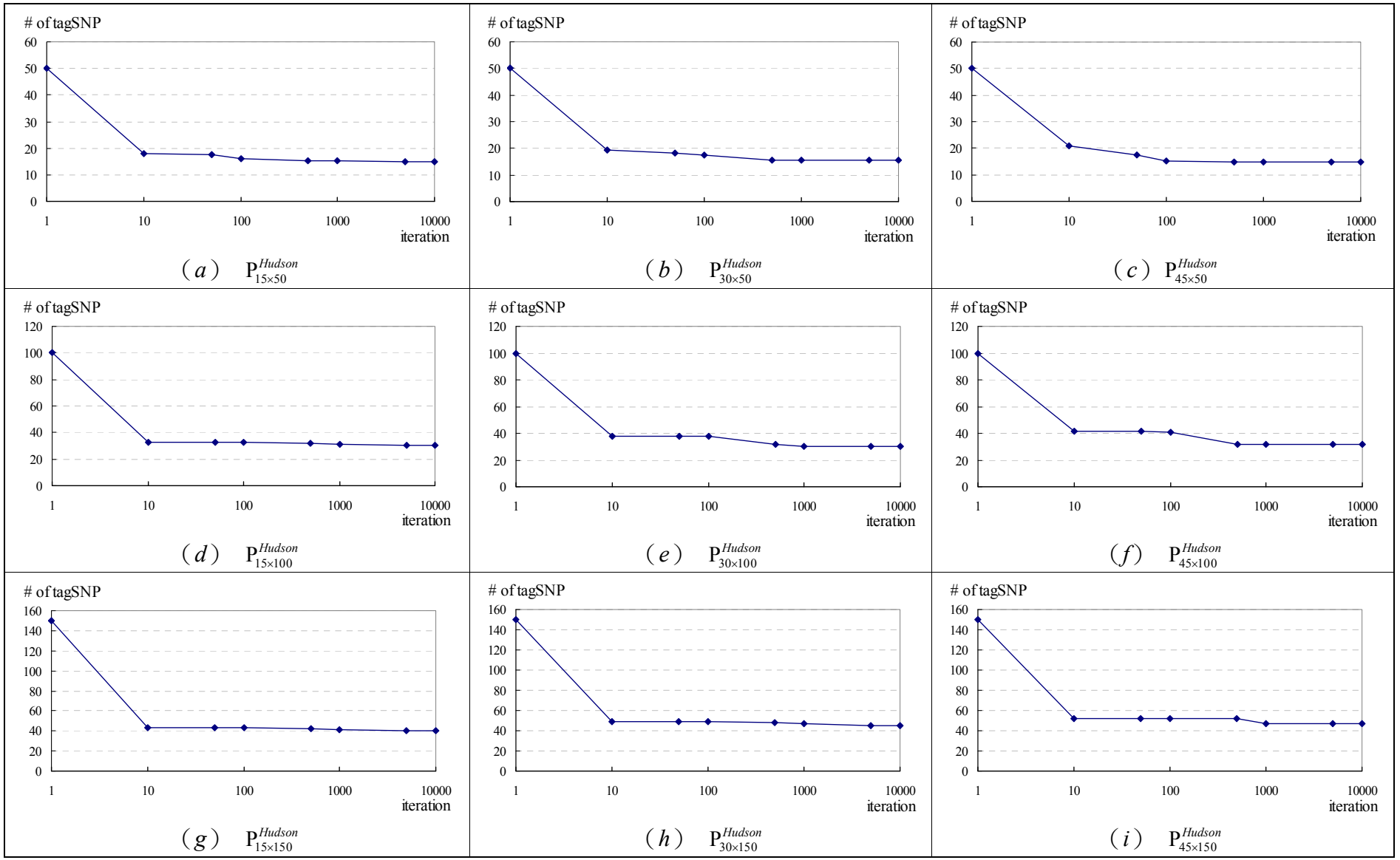


圖 4.8：採用 LRH 求解 Hudson 資料之收斂情況

由於 tagSNP 選取為一個整數規劃問題，其可行解在實數區域中是以跳躍不連續的方式存在，而非連續可微分的區域，因此很難透過採用以連續可微分為基礎之啟發式拉氏鬆弛方式逐漸逼近整數最佳解。此外，我們所提出的啟發式求解過程中有部分步驟強制配對選取某些解，易讓所求之收斂解落入局部解之狀況，因此藉由調整λ或是放鬆強制的配對求解方式必須經由不斷測試模擬才能獲得較好的近似解，如此將耗費不少時間，有違該演算法的設計初衷；是故，以下之數值測試僅將 LRH 之遞迴次數設為 1000 次，而λ與固定欄位規則部分則不做變動。

我們在以 CPLEX 求解時發現該軟體在求解小規模的 tagSNP 選取問題方面十分有效率，可在極短時間內獲得精確最佳解；而 LRH 之特色則是在遞迴初期可快速縮小問題規模，根據 CPLEX 與 LRH 的求解特性，我們提出一個結合 LRH 與 CPLEX 優點之二階段求解方法，稱之為 MIX 演算法。MIX 將先以遞迴次數設定為 100 次的 LRH 求解，接著再將簡化過的問題以 CPLEX 求解之。

以下的測試使用三種演算法：(1) 以 LRH 演算法求解，將遞迴次數設定為 1000 次；(2) 以 MIX 演算法求解；(3) 以 CPLEX 方式求精確解。由於問題模式之主要受變數 m 之大小影響，因此將呈現測試結果的圖表之橫軸以 m 之大小為主排序，n 之大小為次排序（$P_{15\times50} \rightarrow P_{15\times100} \rightarrow P_{15\times150} \rightarrow P_{30\times50} \rightarrow P_{30\times100} \rightarrow P_{30\times150} \rightarrow P_{45\times50} \rightarrow P_{45\times100} \rightarrow P_{45\times150}$）。

圖 4.9 為 CPLEX、LRH 與 MIX 分別在不同規模中平均選取的 tagSNP 總數，從圖中發現不論是何種模擬資料下 LRH 之求解結果與 CPLEX 所求之精確解差距不大。而採用 MIX 的方式求解，更可提升求解的品質與效率，且其改善效果隨問題規模增大而更顯著。

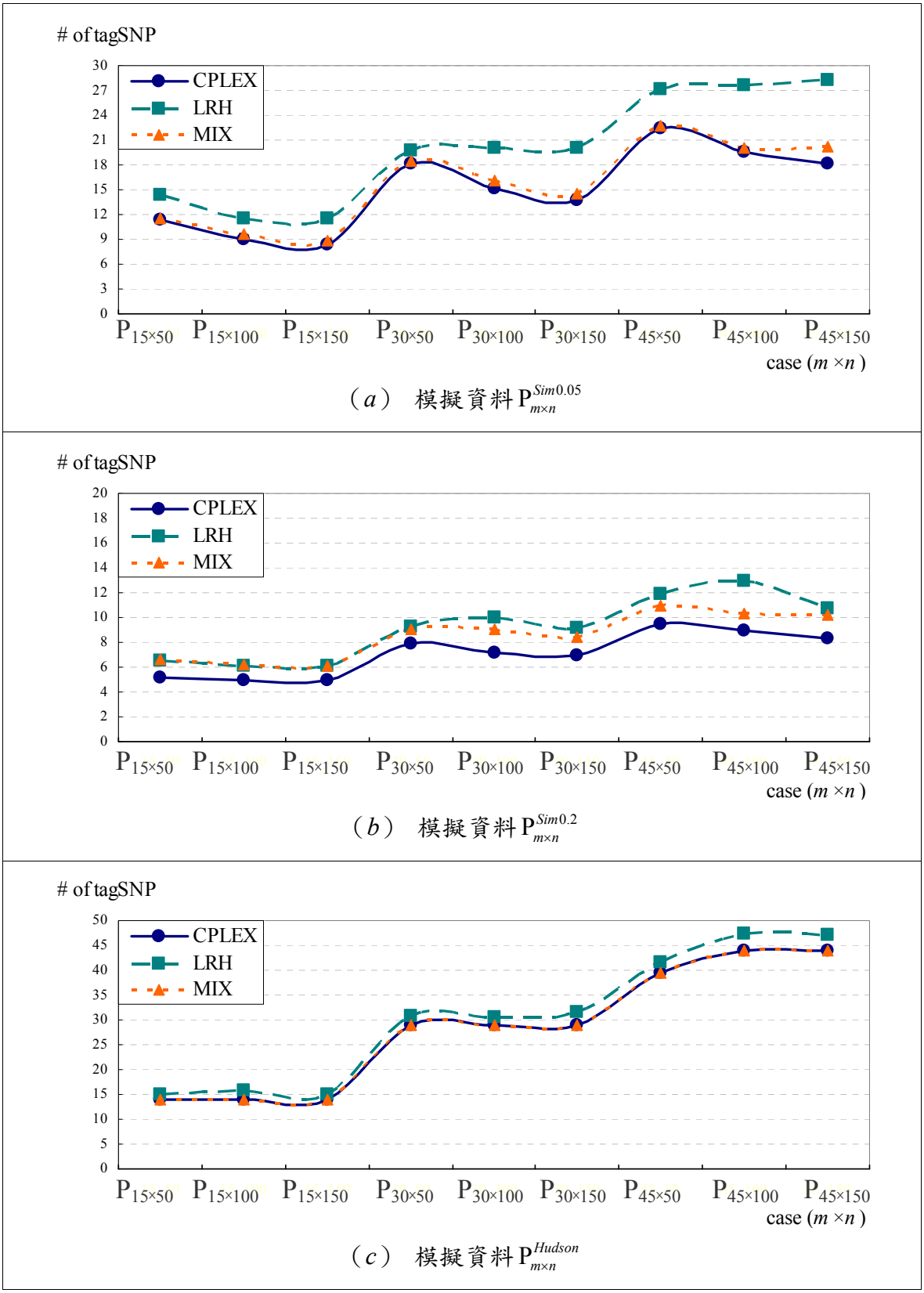


圖 4.9：CPLEX、LRH 與 MIX 之測試結果比較

表 4.1 除列出 CPLEX、LRH 與 MIX 在不同資料來源與不同問題規模中平均選取的 tagSNP 總數之外，並比較 LRH、MIX 之目標函數值與 CPLEX 所解出的最佳目標函數值間的差距（Optimality gap，OPT gap）。OPT gap 之計算方式如下：以α代表方法（LRH 或 MIX）；由 OPT gap 可看出採用 MIX 之方法可大幅度提升求解的精確度。

$$\text{OPT gap}_{\alpha} = \frac{Z(x_{\alpha}) - Z(x_{CPLEX})}{Z(x_{CPLEX})} \times 100\%$$

表 4.1：CPLEX、LRH 與 MIX 之測試結果與 OPT gap

	$P^{Sim0.05}_{15\times50}$	$P^{Sim0.05}_{15\times100}$	$P^{Sim0.05}_{15\times150}$	$P^{Sim0.05}_{30\times50}$	$P^{Sim0.05}_{30\times100}$	$P^{Sim0.05}_{45\times50}$	$P^{Sim0.05}_{30\times150}$	$P^{Sim0.05}_{45\times100}$	$P^{Sim0.05}_{45\times150}$
CPLEX	**11.4**	**9.0**	**8.4**	**18.1**	**15.1**	**13.7**	**22.5**	**19.6**	**18.1**
LRH	**14.3**	**11.5**	**11.6**	**19.8**	**20.0**	**20.1**	**27.2**	**27.7**	**28.3**
（OPT gap）	25.44%	27.78%	38.10%	9.39%	32.45%	46.72%	20.89%	41.33%	56.35%
MIX	**11.5**	**9.7**	**8.9**	**18.4**	**16.1**	**14.6**	**22.7**	**20.1**	**20.2**
（OPT gap）	0.88%	7.78%	5.95%	1.66%	6.62%	6.57%	0.89%	2.55%	11.60%
	$P^{Sim0.2}_{15\times50}$	$P^{Sim0.2}_{15\times100}$	$P^{Sim0.2}_{15\times150}$	$P^{Sim0.2}_{30\times50}$	$P^{Sim0.2}_{30\times100}$	$P^{Sim0.2}_{45\times50}$	$P^{Sim0.2}_{30\times150}$	$P^{Sim0.2}_{45\times100}$	$P^{Sim0.2}_{45\times150}$
CPLEX	**5.2**	**5.0**	**5.0**	**7.9**	**7.2**	**7.0**	**9.5**	**8.9**	**8.3**
LRH	**6.5**	**6.1**	**6.1**	**9.3**	**10.0**	**9.2**	**11.9**	**12.9**	**10.7**
（OPT gap）	25.00%	22.00%	22.00%	17.72%	38.89%	31.43%	25.26%	44.94%	28.92%
MIX	**6.6**	**6.2**	**6.1**	**9.1**	**9.1**	**8.4**	**11.0**	**10.3**	**10.2**
（OPT gap）	26.92%	24.00%	22.00%	15.19%	26.39%	20.00%	15.79%	15.73%	22.89%
	$P^{Hudson}_{15\times50}$	$P^{Hudson}_{15\times100}$	$P^{Hudson}_{15\times150}$	$P^{Hudson}_{30\times50}$	$P^{Hudson}_{30\times100}$	$P^{Hudson}_{45\times50}$	$P^{Hudson}_{30\times150}$	$P^{Hudson}_{45\times100}$	$P^{Hudson}_{45\times150}$
CPLEX	**14.0**	**14.0**	**14.0**	**29.0**	**29.0**	**29.0**	**39.6**	**44.0**	**44.0**
LRH	**15.1**	**15.7**	**14.9**	**30.8**	**30.5**	**31.5**	**41.5**	**47.5**	**47.2**
（OPT gap）	7.86%	12.14%	6.43%	6.21%	5.17%	8.62%	4.80%	7.95%	7.27%
MIX	**14.0**	**14.0**	**14.0**	**29.0**	**29.0**	**29.0**	**39.6**	**44.0**	**44.0**
（OPT gap）	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

由圖 4.10 中的執行時間可得以下觀察：CPLEX 在求解 Hudson 模擬資料時不受問題規模影響皆可快速獲得精確解；但在求解 Sim0.05 與 Sim0.2 之模擬資料時，其求解時間隨問題規模變大而快速上升；而 LRH 之執行時間則較不受資料種類與問題規模影響，皆可快速獲得一近似解；MIX 方法結合了 LRH 與 CPLEX 兩種解法的優點，可獲得高品質與高效率之結果。

表 4.2 將執行時間以倍率方式呈現，可幫助觀察三種方法其執行時間上的差距。其中 MIX 之執行時間在大部分規模的問題中皆為最小，因此以它作標準化時間（Normalized Time，NT）的基礎。其標準化式子如下，其中α代表方法（CPLEX 或 LRH）；從表 4.2 中的 NT 值亦可看出 CPLEX 在求解大規模問題時其花費時間呈跳躍式成長。

$$NT_{\alpha} = \frac{T_{\alpha}}{T_{MIX}}$$

接下來分析 LRH 與 MIX 在每種問題規模下 10 個不同案例與最佳解間的差距比較。圖 4.9、圖 4.10、圖 4.11 分別為三種不同資料種類其各規模中 10 個案例的最大誤差、平均誤差及最小誤差的分佈情形，並將對應的值以表 4.3 列出，從中可發現採用 MIX 之求解方法可大幅度提升求解品質；值得注意的是，在模擬 Hudson 資料時，MIX 所求結果與 CPLEX 精確解完全相同；而單以 LRH 求解之結果可能因所測試的資料種類不同而有差異，譬如 Sim0.05 資料的求解結果與精確解間的差異易受測試案例不同影響。

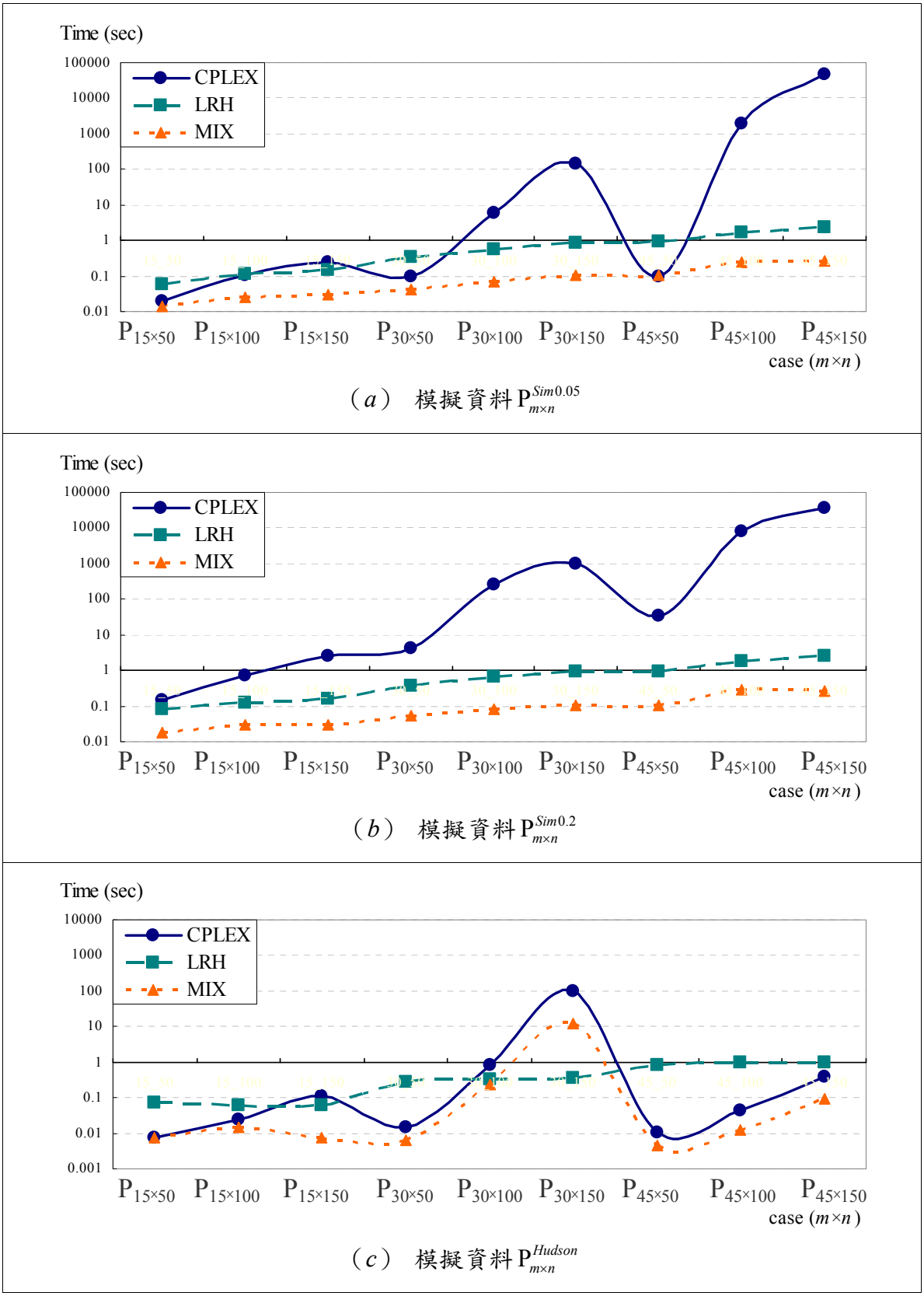
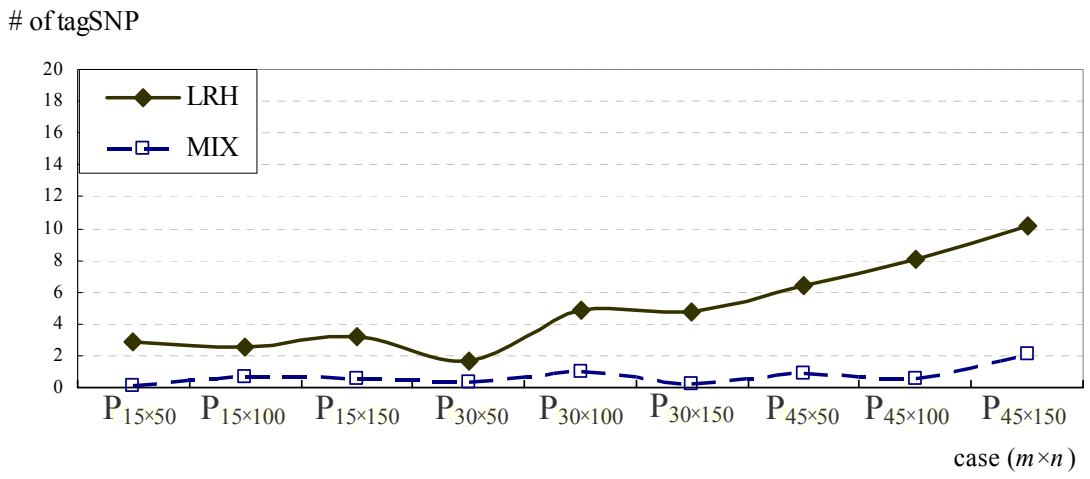


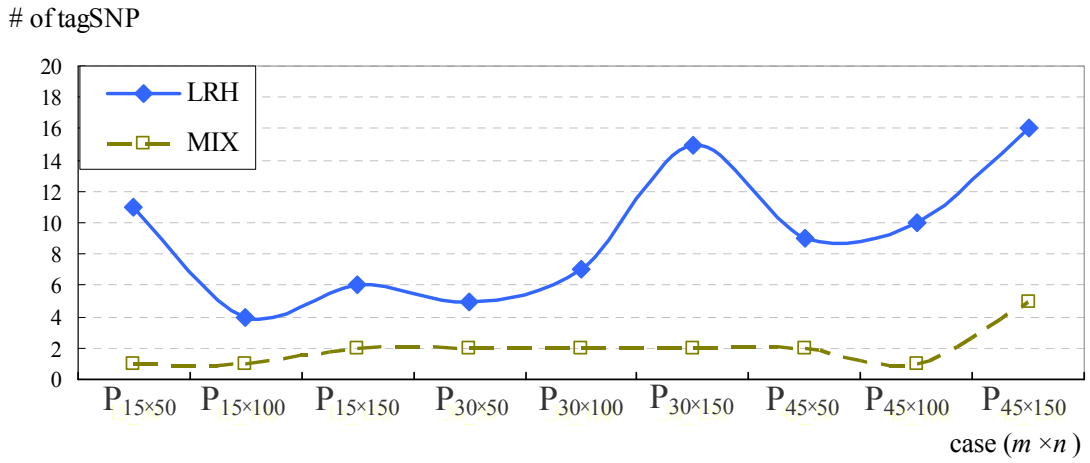
圖 4.10：CPLEX、LRH 與 MIX 之測試時間比較

表 4.2：CPLEX、LRH 與 MIX 之測試實際時間與標準化時間（NT）

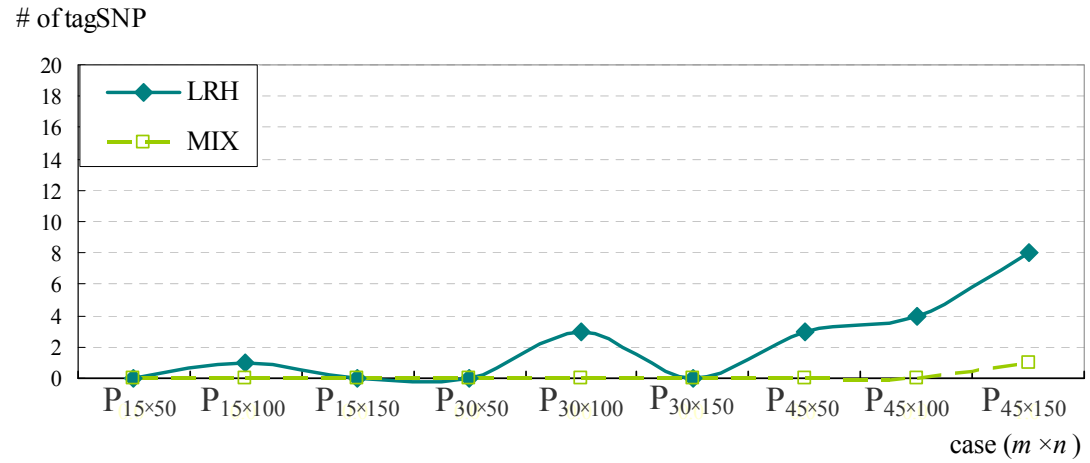
	$P^{Sim0.05}_{15\times50}$	$P^{Sim0.05}_{15\times100}$	$P^{Sim0.05}_{15\times150}$	$P^{Sim0.05}_{30\times50}$	$P^{Sim0.05}_{30\times100}$	$P^{Sim0.05}_{45\times50}$	$P^{Sim0.05}_{30\times150}$	$P^{Sim0.05}_{45\times100}$	$P^{Sim0.05}_{45\times150}$
CPLEX	**0.02**	**0.11**	**0.25**	**0.10**	**5.89**	**138.45**	**0.10**	**1908.91**	**47613.31**
（NT）	1.49	4.41	8.30	2.24	87.98	1292.69	0.94	7611.28	176280.31
LRH	**0.06**	**0.11**	**0.14**	**0.34**	**0.57**	**0.87**	**0.90**	**1.70**	**2.44**
（NT）	4.21	4.51	4.70	7.86	8.59	8.09	8.52	6.78	9.02
MIX	**0.01**	**0.02**	**0.03**	**0.04**	**0.07**	**0.11**	**0.11**	**0.25**	**0.27**
	$P^{Sim0.2}_{15\times50}$	$P^{Sim0.2}_{15\times100}$	$P^{Sim0.2}_{15\times150}$	$P^{Sim0.2}_{30\times50}$	$P^{Sim0.2}_{30\times100}$	$P^{Sim0.2}_{45\times50}$	$P^{Sim0.2}_{30\times150}$	$P^{Sim0.2}_{45\times100}$	$P^{Sim0.2}_{45\times150}$
CPLEX	**0.15**	**0.75**	**2.47**	**4.22**	**254.48**	**1027.64**	**35.43**	**8330.64**	**36438.52**
（NT）	8.31	25.84	84.70	80.60	3084.60	9419.24	329.87	30150.71	136422.77
LRH	**0.08**	**0.12**	**0.16**	**0.37**	**0.66**	**0.90**	**0.95**	**1.76**	**2.46**
（NT）	4.54	4.16	5.45	6.99	8.03	8.25	8.86	6.38	9.22
MIX	**0.02**	**0.03**	**0.03**	**0.05**	**0.08**	**0.11**	**0.11**	**0.28**	**0.27**
	$P^{Hudson}_{15\times50}$	$P^{Hudson}_{15\times100}$	$P^{Hudson}_{15\times150}$	$P^{Hudson}_{30\times50}$	$P^{Hudson}_{30\times100}$	$P^{Hudson}_{45\times50}$	$P^{Hudson}_{30\times150}$	$P^{Hudson}_{45\times100}$	$P^{Hudson}_{45\times150}$
CPLEX	**0.01**	**0.03**	**0.11**	**0.01**	**0.82**	**99.91**	**0.01**	**0.04**	**0.39**
（NT）	1.01	1.66	14.59	2.31	3.47	8.27	2.33	3.41	4.08
LRH	**0.07**	**0.06**	**0.06**	**0.29**	**0.32**	**0.36**	**0.84**	**0.94**	**1.02**
（NT）	9.61	3.89	7.95	47.07	1.38	0.03	186.73	76.69	10.72
MIX	**0.01**	**0.02**	**0.01**	**0.01**	**0.24**	**12.07**	**0.00**	**0.01**	**0.09**



（*a*） LRH、MIX 與精確解之平均誤差比較

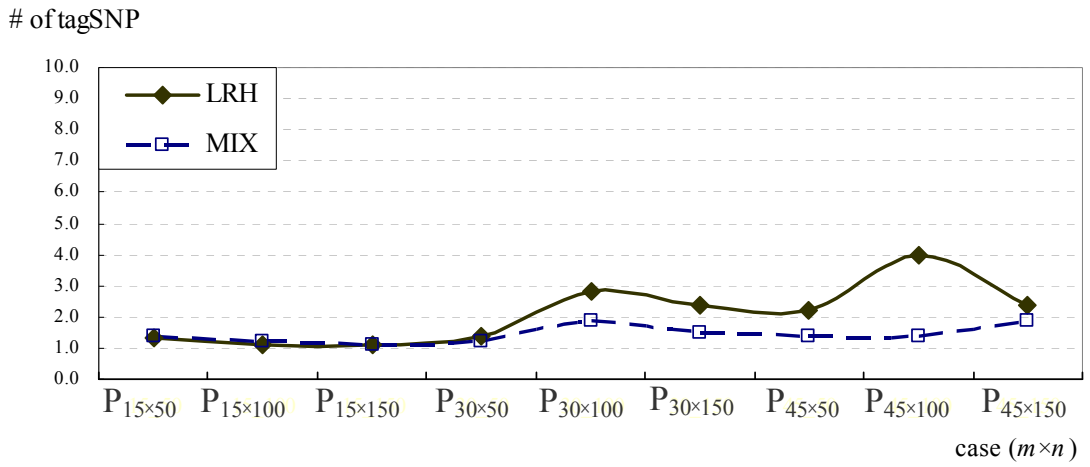


（*b*） LRH、MIX 與精確解最大誤差之比較

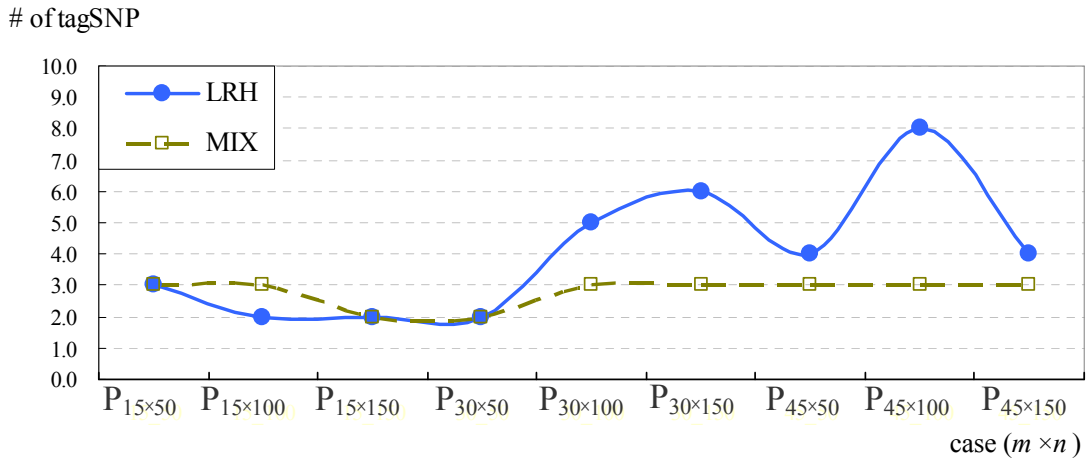


（*c*） LRH、MIX 與精確解最小誤差之比較

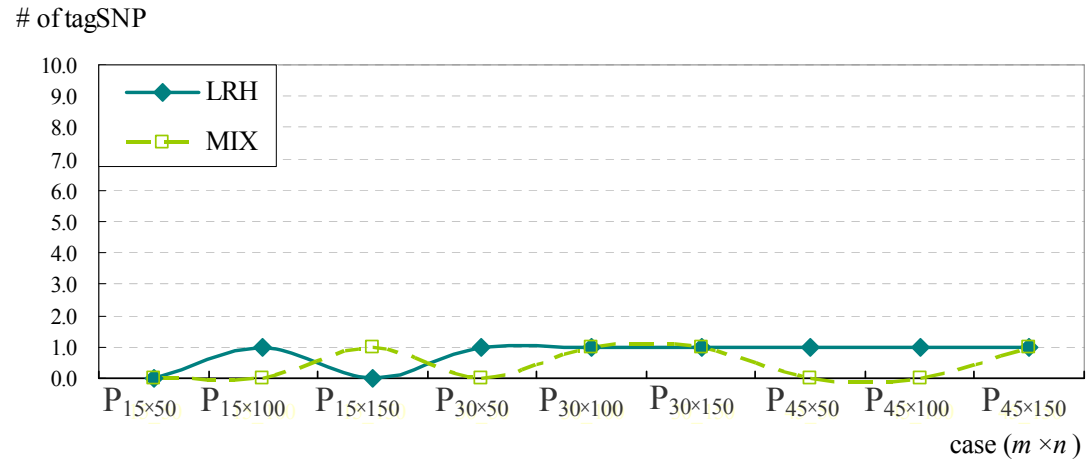
圖 4.11：Sim0.05 資料之求解結果



（*a*） LRH、MIX 與精確解之平均誤差比較

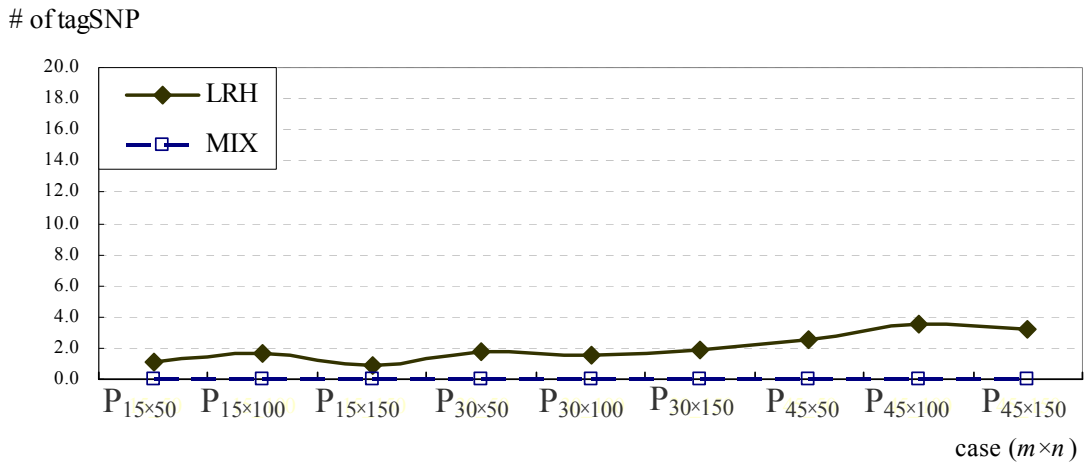


（*b*） LRH、MIX 與精確解最大誤差之比較

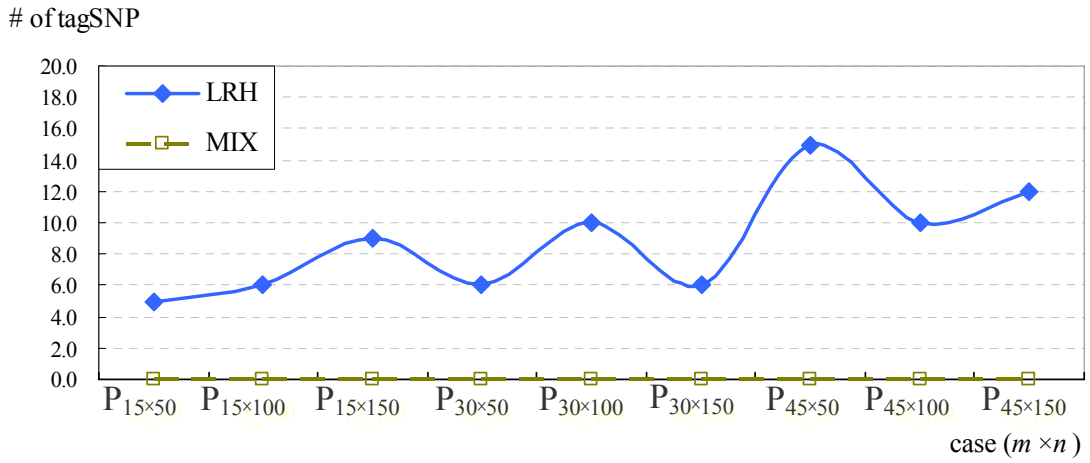


（*c*） LRH、MIX 與精確解最小誤差之比較

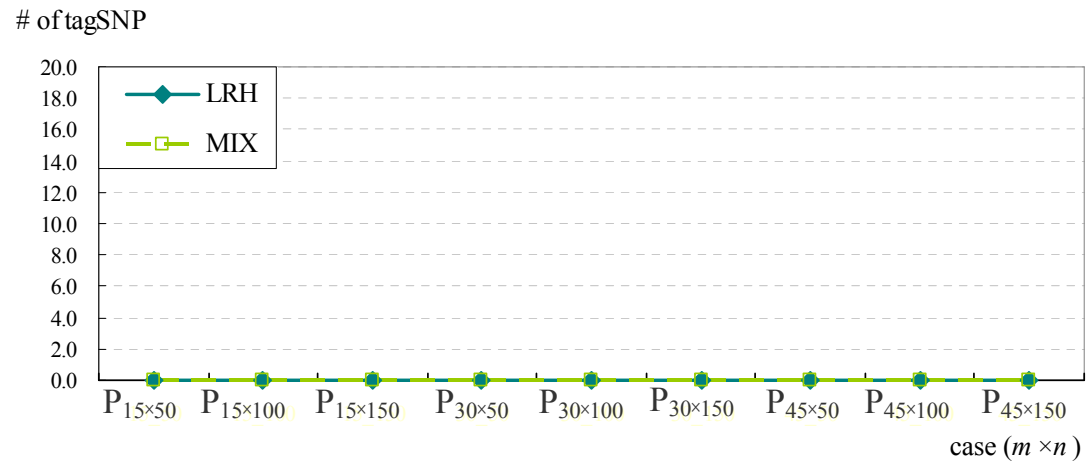
圖 4.12：Sim0.2 資料之求解結果



（*a*） LRH、MIX 與精確解之平均誤差比較



（*b*） LRH、MIX 與精確解最大誤差之比較



（*c*） LRH、MIX 與精確解最小誤差之比較

圖 4.13：Hudson 資料之求解結果

表 4.3：LRH、MIX 與精確解之最大誤差、最小誤差及平均誤差

		$P_{15\times50}$	$P_{15\times100}$	$P_{15\times150}$	$P_{30\times50}$	$P_{30\times100}$	$P_{45\times50}$	$P_{30\times150}$	$P_{45\times100}$	$P_{45\times150}$
LRH	Max	11.0	4.0	6.0	5.0	7.0	15.0	9.0	10.0	16.0
	Min	0.0	1.0	0.0	0.0	3.0	0.0	3.0	4.0	8.0
	Avg.	2.9	2.5	3.2	1.7	4.9	4.7	6.4	8.1	10.2
MIX	Max	1.0	1.0	2.0	2.0	2.0	2.0	2.0	1.0	5.0
	Min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
	Avg.	0.1	0.7	0.5	0.3	1.0	0.2	0.9	0.5	2.1
		$P_{15\times50}$	$P_{15\times100}$	$P_{15\times150}$	$P_{30\times50}$	$P_{30\times100}$	$P_{45\times50}$	$P_{30\times150}$	$P_{45\times100}$	$P_{45\times150}$
LRH	Max	3.0	2.0	2.0	2.0	5.0	6.0	4.0	8.0	4.0
	Min	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
	Avg.	1.3	1.1	1.1	1.4	2.8	2.4	2.2	4.0	2.4
MIX	Max	3.0	3.0	2.0	2.0	3.0	3.0	3.0	3.0	3.0
	Min	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0
	Avg.	1.4	1.2	1.1	1.2	1.9	1.5	1.4	1.4	1.9
		$P_{15\times50}$	$P_{15\times100}$	$P_{15\times150}$	$P_{30\times50}$	$P_{30\times100}$	$P_{45\times50}$	$P_{30\times150}$	$P_{45\times100}$	$P_{45\times150}$
LRH	Max	5.0	6.0	9.0	6.0	10.0	6.0	15.0	10.0	12.0
	Min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Avg.	1.1	1.7	0.9	1.8	1.5	1.9	2.5	3.5	3.2
MIX	Max	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Avg.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

4.3.2.2 真實生物資料之測試結果

由於真實資料中可能遺漏某些 SNP 資訊，我們以圖 4.14 說明如何處理遺漏資訊的處理方式。首先將所有的 SNP 資訊轉為矩陣資料，以 2 表示遺漏的 SNP 資訊，因此可得到如圖 4.14 中的矩陣 H。在轉為辨識矩陣 ES 時，如欲得知 S_2 可辨識的 Haplotype pair，僅在與鹼基 S_2 交集之 Haplotype pair $\left(h_{i_1,j_1}, h_{i_2,j_2}\right)$ 為 $(0,1)$ 與 $(1,0)$ 時，其對應的 indicator function $I\left(E_{i_1,i_2}, S_2\right)=1$ 表示可辨識；其他情形則表示不能辨識，其對應之 $I\left(E_{i_1,i_2}, S_2\right)=0$。舉例來說，對圖 4.14 中矩陣 H 而言，若欲以 S_2 欄位辨識 h_1 與 h_2，

由於$(h_{1,2},h_{2,2})$恰為$(0,1)$，因此其對應之$I(E_{1,2},S_2)=1$；反之，若欲以S_2欄位辨識h_1與h_3，由於$(h_{1,2},h_{3,2})$為$(0,2)$，而非$(0,1)$、$(1,0)$兩者其中之一，因此其對應之$I(E_{1,3},S_2)=0$。以此類推，可得圖 4.14 中的矩陣 *ES*。

$$H=\begin{array}{c} \\ h_1\\ h_2\\ h_3\\ h_4\end{array}\begin{array}{c}\begin{array}{ccccc} S_1 & S_2 & S_3 & S_4 & S_5\end{array}\\ \begin{bmatrix}1&0&0&1&1\\0&1&0&1&2\\1&2&1&0&0\\1&1&1&2&1\end{bmatrix}\end{array} \Longrightarrow ES=\begin{array}{c} \\ E_{1,2}\\ E_{1,3}\\ E_{1,4}\\ E_{2,3}\\ E_{2,4}\\ E_{3,4}\end{array}\begin{array}{c}\begin{array}{ccccc} S_1 & S_2 & S_3 & S_4 & S_5\end{array}\\ \begin{bmatrix}1&1&0&0&0\\0&1&1&1&1\\0&0&1&0&0\\1&0&1&1&0\\1&0&1&0&0\\0&0&0&0&1\end{bmatrix}\end{array}$$

$$\begin{array}{c} h_1\\ h_2\\ h_3\\ h_4\end{array}\begin{bmatrix}0\\1\\2\\1\end{bmatrix}\begin{matrix}E_{1,2}\\E_{2,3}\\E_{3,4}\end{matrix}\Big)\Big)\Big)E_{1,3}\Big)E_{2,4}\Big)E_{1,4} \Rightarrow \begin{cases}(h_{1,2},h_{2,2})=(0,1)\Rightarrow I(E_{1,2},S_2)=1,\ (h_{1,2},h_{3,2})=(0,2)\Rightarrow I(E_{1,3},S_2)=0,\\ (h_{1,2},h_{4,2})=(0,1)\Rightarrow I(E_{1,4},S_2)=1,\ (h_{2,2},h_{3,2})=(1,2)\Rightarrow I(E_{2,3},S_2)=0,\\ (h_{2,2},h_{4,2})=(2,1)\Rightarrow I(E_{2,4},S_2)=0,\ (h_{3,2},h_{4,2})=(1,1)\Rightarrow I(E_{3,4},S_2)=0,\end{cases}$$

圖 4.14：真實資料中遺漏資訊的處理方式示意圖

將 Patil et al.（2001）測試的 20 個不同個體的第 21 號染色體上 24047 個 SNP 依上述方式產生辨識矩陣 *ES*，如此便可轉換成數學模式求解，以下將真實資料的測試分為兩個部份。

第一個部份針對已切割 Block 後的小規模問題進行求解，此部分採用 Patil et al.（2001）切割後的 Block 資訊，共有 4135 個 Block；針對其中每一個 Block 分別以 LRH 與 CPLEX 進行求解，最後加總所有已選取的 tagSNP 個數。由於 Patil et al.所選取之 tagSNP 僅需達到辨識出每一個 Block 中出現次數 80%的 Haplotype pattern 即可，與本論文所定義的 tagSNP（達到完整辨識所有 Haplotype 樣式之差異）有些許差異，因此不宜與本論文所求得之結果比較。另一個部份對未切割 Block 的大規模問題進行求解，由於問題規模過大，CPLEX 耗費太多時間仍無法解完，因此我們僅採用 LRH 演算法進行求解。

測試結果顯示 LRH 求解 4135 個 Block 總共需要 9211 個 tagSNP，比 CPLEX 所

求之精確解 8813 個 tagSNP 還要多 398 個;然而,LRH 對於原 4135 個 Block 中的 3821 個 Block 所求之結果與精確解完全相同，而剩餘的 314 個 Block 之求解結果平均比精確值多了 1.27 個，平均的 OPT gap 為 3.29%。倘若採用 LRH 直接求解未切割 Block 之問題，則僅需要 10 個 tagSNP 即可達到完整辨識之目的，此結果與經過切割 Block 後精確解所需要的 8813 個 tagSNP 相差頗大，其原因在於總共須被辨識的 Haplotype 樣式在切割 Block 後將由原先未切割 Block 前的 20 個不同樣式變為 14234 個不同樣式，而欲達到辨識這些增加的樣式則需再增加選取的 tagSNP 個數。在生物應用上，何種結果（即切或不切 Block）較佳則須視其使用目的不同而有不同。若僅需達到辨識 20 個不同個體的目的,則不切 Block 而選取 10 個 tagSNP 的方式會是較佳的選擇，因為如此可增加晶片的剩餘容量供其他應用目的使用，譬如另加入其他 tagSNP 的選取用來作 SNP 序列的延伸預測等等；反之，倘若所切割的 Block 有其特殊意義的話，則切 Block 再從中選取 8813 個 tagSNP 將會是比較好的選擇。

4.4 小結

本章節提出一個 LRH 演算法求解 tagSNP 選取問題，其與一般求解集合涵蓋問題的拉氏鬆弛法最大差異在於 LRH 加入的修正法則，透過修正後的演算法可使求解品質及效果更好。根據測試的結果顯示 LRH 之求解速率良好，且求解品質亦不差，十分適合用於大規模的選取問題上；另外，由於 LRH 最大的優點在於遞迴次數的初期可快速獲得一個近似解，而 CPLEX 線性規劃軟體在求解小規模整數規劃問題時十分有效率，加上若採用在求解的過程中不斷微調幅度參數的方式來提升 LRH 的求解品質的話，其成效將受使用者經驗影響且將可能耗費許多測試時間，不符合原求解演算法之設計初衷，因此本論文亦結合 LRH 與 CPLEX 軟體之優點，提出另一個二階段的 MIX 求解方法作測試。MIX 演算法先用 LRH 快速獲得一組具不錯可行解的縮

小版 tagSNP 選取問題，再用 CPLEX 求得其精確解。測試數據顯示 MIX 演算法不僅十分有效率，且可大幅提升求解的品質。最後，我們亦提出演算法之修改步驟以解決在發生資訊遺漏情況時如何求解 tagSNP 選取問題。

第五章 結論與未來研究方向

本章首先總結整篇論文並列出具體貢獻，接著在 5.2 小節建議一些後續可能的研究主題與方向，其中包括針對具容量限制之生物晶片（Biochip）上應選取那些 tagSNP 以達到更可靠（reliable）或強健（robust）的目標，以及其它與 tagSNP 相關的研究問題或求解方式。

5.1 論文總結與貢獻

本研究主要目的為選取具代表性的單核苷酸多型性以有效地減少原本龐大的生物資訊儲存空間以利後續研究應用。在第三章中我們提出一個以圖形理論為基礎之 Multi-TagSNP 演算法，用以求出 tagSNP 選取問題之多重最佳解，並加入連鎖不平衡觀念以從這些多重最佳解中進一步挑選出一組更具生物資訊意義的最佳解。

而後，本研究亦提出一個可同時考慮極小化所選取之 tagSNP 個數以及極大化該 tagSNP 與其它 SNP 間 LD 值之和的雙目標整數規劃模式。透過該數學模式決策者可依據自己的偏好針對不同的目標給予不同的效用函數以求解此雙目標整數規劃問題。由於此類 tagSNP 選取問題一般會依使用目的之不同而有不同的模式表現型態，若能同時考慮多個不同目標，則可增加其使用範圍或目的。

本研究之第四章中用拉氏鬆弛法的概念發展一個啟發式求解演算法 LRH 以在求解大規模問題時可快速獲得一個收收斂解，雖此收斂解不保證為最佳解，但其求解效率十分良好，求解品質亦不算太差。由於觀察到 CPLEX 最佳化規劃軟體在求解小規模的 tagSNP 選取問題十分有效率，因此我們發展出一個兩階段的 MIX 演算法，其第一階段先用 LRH 求解大規模的 tagSNP 選取問題，選出較好的可能候選 SNP 之後，第二階段再以這些候選 SNP 為基礎建構一較小規模的 tagSNP 選取問題並以 CPLEX 求解之。由多項測試結果顯示 MIX 演算法成效極

高，可在短時間內獲得品質極高之解。茲將本論文的具體貢獻歸納如下：

1. 提出一個求解 tagSNP 選取問題之多重最佳解的演算法 Multi-TagSNP。
2. 提出在選取 tagSNP 問題中同時極大化 LD 值之和之雙目標數學規劃模式。
3. 提出以啟發式拉氏鬆弛法 LRH 求解大規模之 tagSNP 選取問題。
4. 修正拉氏鬆弛法中選取對應解與與固定欄位之規則以增進演算效率。
5. 提出結合 LRH 與 CPLEX 之二階段演算法 MIX 以獲得高效率及高品質之求解結果。
6. 針對具容量限制之 Biochip 上 tagSNP 選取問題提出一個可選取較可靠或強健的 tagSNP 數學規劃模式。

5.2 未來研究方向

5.2.1 具容量限制之生物晶片上選取較可靠的 tagSNP 問題

由於生物晶片在執行試驗時可能會失敗，如此可能導致某一 SNP 無法辨識該 SNP 原本可辨識的 Haplotype pair 之情況，因此倘若要求所選取之 tagSNP 僅可將每一 Haplotype pair 辨識至少僅「一次」的話，則當某些試驗發生失敗時，可能造成某些 Haplotype pair 無法被辨識出來。舉例來說，假使圖 5.1 左邊之 Haplotype matrix 因試驗失敗而導致部分資訊遺漏（即那些標有？之 cell），則所選取出的 tagSNP（S_4 及 S_6）將無法辨識 h_i 究竟是 h_2 或是 h_4。

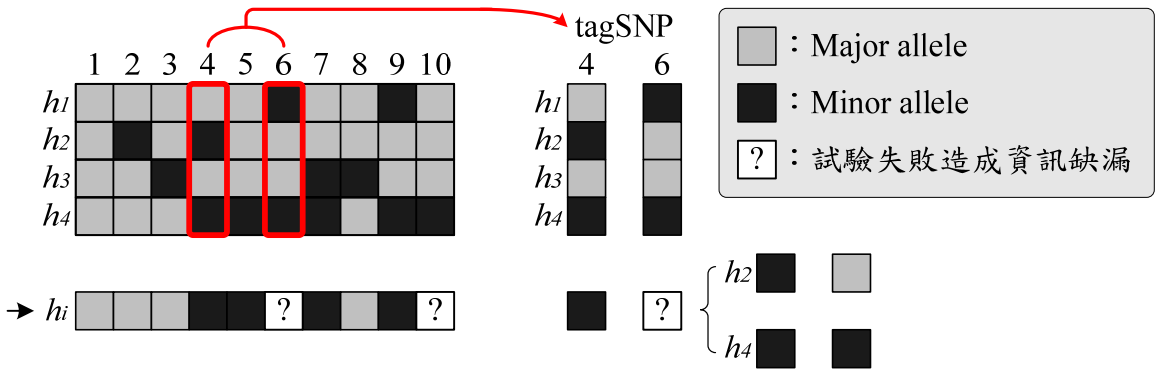


圖 5.1：遺漏資訊對辨識 Haplotype pair 之影響

在此我們先定義每一 Haplotype pair 被辨識的次數限制為 F，則第三章所定義的 tagSNP 選取問題之 F=1。而當辨識次數 F 設為 2 時，代表每一個 Haplotype pair 至少可被 2 個 SNP 辨識，此時倘若其中一個 SNP 因試驗失敗而遺漏資訊，則仍可透過另一個 SNP 以達到辨識該 Haplotype pair 的目的。因此，增加 Haplotype pair 被辨識的次數限制 F 可以使該組被選取的 tagSNP 應用於辨識上更為可靠。

雖然增加 F 值可能會增加 tagSNP 的選取個數，然而生物晶片上可容納的 SNP 個數上限 C 在選取 tagSNP 前即可得知，因此我們可以將 F 與 C 皆納入以下列出之數學規劃模式 P^{C}_{tagSNP} 中。模式 P^{C}_{tagSNP} 為考慮容量 C 之限制下，所選取的 tagSNP 可將每一個 Haplotype pair 至少辨識 F 次，而目標在於極大化 F。

$$
\begin{aligned}
&\mathit{Max} \quad F && (\mathrm{P}^{C}_{tagSNP})\\
&\mathit{st.} \quad \sum\nolimits_{S_j \in \mathrm{S}} x_j \le C\\
&\qquad \sum\nolimits_{(S_j, E_{i_1,i_2}) \in A} x_j \ge F \ , \ \forall E_{i_1,i_2} \in \mathrm{E}\\
&\qquad x_j \in \{0,1\}, \ \forall S_j \in \mathrm{S}
\end{aligned}
$$

由於生物晶片容量 C 之大小應視實際需求而取決於製造者所花費的成本與技術限制，因此我們以下之數值測試僅呈現 F 與 C'之間的關係，其中 C'代表容量 C 的下界，亦即在 F 之限制下最少需選取的 tagSNP 個數。而晶片製造者可藉由 F 與 C'之間的關係，決定在考慮技術與成本的情況下，選取對其收益最高的 tagSNP。

由於在求解 Hudson 資料時，的情況幾乎所有的案例皆無可行解（亦即大部分 Hudson 的資料僅適用 $F \in \{1,2\}$ 的情形），因此下列 F>2 的數值分析測試將僅呈現模擬資料 Sim0.05 與 Sim0.2 之求解結果。圖 5.2 與圖 5.3 分別為測試資料 Sim0.05 與資料 Sim0.2 時，各種問題規模下各取一個範例所需的 tagSNP 數。從圖中確實可觀察出當 F 值設定越高時 C'值亦隨之越高，代表欲涵蓋越多次 Haplotype pair 時所需之晶片容量越大。晶片製造者可藉由 F 與 C'之間的關係，

決定在考慮技術與成本的情況下，對其收益最高的決策方式。

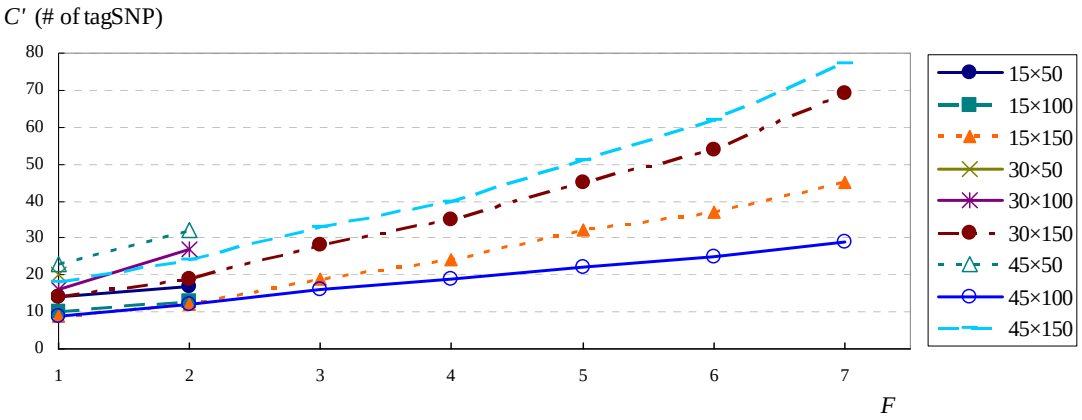


圖 5.2：Sim0.05 資料之 *F* 與 *C'*關係圖

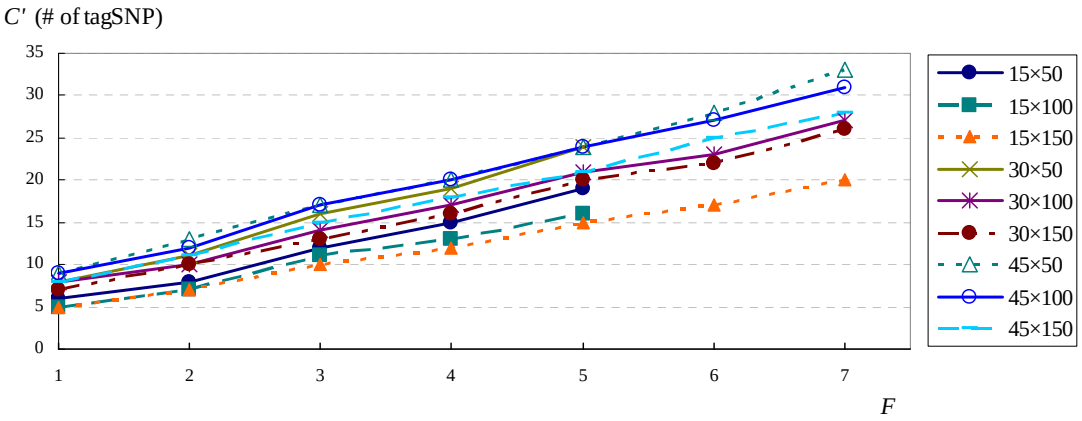


圖 5.3：Sim0.2 資料之 *F* 與 *C'*關係圖

5.2.2 其它之研究方向建議

從生物上的角度來看，當兩個基因座相鄰越緊密，其重組率越小，連鎖不平衡的情形也越明顯，因此可考慮將兩個 SNP 在 Haplotype 上相距的長度當作另一種多重 tagSNP 評準方法，或是加入其他更具生物意義的條件作為新的評選方法。

此外，或可試著使用其它的最佳化方法求解此 tagSNP 選取問題（亦即，集合涵蓋問題）。舉例來說，Balas 於 1971 年所提出的切面法（branch and cut）在

求解整數規劃問題的過程中找到一些限制法則，以增加限制式的方式不斷縮減可行解區域，加快求解時間。而求解放鬆後的拉氏問題時，或可改用如 Bundle Method（Frangioni and Gallo，1999）等類似次梯度法的方式求解。

次梯度法中亦有不少改良的空間值得嘗試，譬如我們或可嘗試將第四章作法中強制固定某些欄位的作法加以修正，使原本已選定之欄位有被解除強制固定欄位的彈性，如此可讓求解結果跳脫局部解。例如從檢驗式中可看出每一 Haplotype pair 被涵蓋之次數，可針對該次數設定一懲罰權數以使當懲罰權數超過某一門檻時則釋放某些已選取之欄位，如此可減少因固定欄位而讓所求結果被困於某一局部最佳解的情形發生。

最後，我們或可以 5.2.1 小節所提出的數學模式為基礎，再進一步加上另一具生物意義的指標（譬如第三章的 LD 值等），以決定在晶片容量可接受的程度內該如何選取 tagSNP 才可同時擁有最多原始序列的資訊且所得之解更具可靠（reliability）與強健性（robustness）；以上這些想法皆可使 tagSNP 在生技業的應用範圍更廣。

參考文獻

Al-Sultan, K. S., M. F. Hussain, et al. (1996). "A Genetic Algorithm for the Set Covering Problem." Journal of the Operational Research Society, **47**(5): 702-709.

Arnheim, N., P. Calabrese, et al. (2003). "Hot and Cold Spots of Recombination in the Human Genome: the Reason We Should Find Them and How This Can Be Achieved " American Journal of Human Genetics, **73**(1): 5-16.

Avi-Itzhak, H., X. Su, et al. (2003). Selection of minimum subsets of single nucleotide polymorphisms to capture haplotype block diversity. In Processding of Pacific Symposium on Biocomputing, Lihue, HI.

Bafna, V., B. V. Halld´orsson, et al. (2003). Haplotypes and informative SNP selection algorithms: don't block out information. Proceedings of the seventh annual international conference on Research in computational molecular biology(RECOMB), Berlin, Germany, ACM.

Balas, E. (1971). "Intersection cuts - A new type of cutting planes for integer programming." Operations Research, **19**(1): 19-39.

Beasley, J. E. and P. C. Chu (1996). "A genetic algorithm for the set covering problem." European Journal of Operational Research, **94**(2): 392-404.

Caprara, A., M. Fischetti, et al. (1999). "A Heuristic Method for the Set Covering Problem." Operations Research, **47**(5): 730-743.

Carlson, C. S., M. A. Eberle, et al. (2004). "Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium." American Journal of Human Genetics, **74**(1): 106-120.

Ceria, S., P. Nobili, et al. (1998). "A Lagrangian-based heuristic for large-scale set cover-

ing problems." Mathematical Programming, **81**(2): 215-228.

Chvatal, V. (1979). "A Greedy Heuristic for the set covering problem." Mathematics of Operations Research, **4**(3): 233-235.

Cohon, J. L. (1978). Multiobjective Programming and Planning. New York: Academic.

Devlin, B. and N. Risch (1995). "A Comparison of Linkage Disequilibrium Measures for Fine-Scale Mapping." Genomics, **29**(2): 311-322.

Feo, T. A. and M. G. C. Resende (1989). "A Probabilidtic heuristic for a computationally difficult set covering problem." Operational Research Letters, **8**: 67-71.

Frangioni, A. and Gallo, G. (1999). "A bundle type dual-ascent approach to linear multi-commodity min cost flow problems." INFORMS Journal on Computing, **11**(4): 370–393

Gabriel, S. B., S. F. Schaffner, et al. (2002). "The Structure of Haplotype Blocks in the Human Genome." Science, **296**(5576): 2225-2229.

Garey, M. R. and D. S. Johnson (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, W. H. Freeman.

Gomory, P. C. G. a. R. E. (1961). "A Linear Programming Approach to the Cutting Stock Problem-Part I." Operations Research, **11**: 849-859.

Gomory, P. C. G. a. R. E. (1963). "A Linear Programming Approach to the Cutting Stock Problem-Part II." Operations Research, **11**(6): 863-888.

Haddadi, S. (1997). "Simple Lagrangian heuristic for the set covering problem." European Journal of Operational Research, **97**: 200-204.

Halperin, E., G. Kimmel, et al. (2005). "Tag SNP selection in genotype data for maximizing SNP prediction accuracy." Bioinformatics, **21**(suppl_1): i195-203.

Hifi, M., V. T. Paschos, et al. (2000). "A neural network for the minimum set covering problem." Chaos, Solitons & Fractals, **11**(13): 2079-2089.

Huang, Y.-T., K. Zhang, et al. (2005). "Selecting additional tag SNPs for tolerating missing data in genotyping." BMC Bioinformatics, **6**(14712105): 263.

Johnson, G. C. L., L. Esposito, et al. (2001). "Haplotype tagging for the identification of common disease genes." Nat Genet, **29**(2): 233-237.

Ke, X. and L. R. Cardon (2003). "Efficient selective screening of haplotype tag SNPs." Bioinformatics, **19**(2): 287-288.

Li, W.-H. and D. Graur (1990). Fundamentals of molecular evolution, Sinauer Associates, Inc. Sunderland. Massachusetts.

Lin, C. H. (2005). Linkage Disequilibrium Measure. Department of Statistics. Los Angeles, University of California.

Nowotny, P., J. M. Kwon, et al. (2001). "SNP analysis to dissect human traits." Current Opinion in Neurobiology, **11**(5): 637-641.

Ohlsson, M., C. Peterson, et al. (2001). "An efficient mean field approach to the set covering problem." European Journal of Operational Research, **133**(3): 583-595.

Patil, N., A. J. Berno, et al. (2001). "Blocks of Limited Haplotype Diversity Revealed by High-Resolution Scanning of Human Chromosome 21." Science, **294**(5547): 1719-1723.

Phillips, M. S., R. Lawrence, et al. (2003). "Chromosome-wide distribution of haplotype blocks and the role of recombination hot spots." Nat Genet, **33**(3): 382-387.

Shastry, B. S. (2002). "SNP alleles in human disease and evolution." Journal of Human Genetics, **47**(11): 561-566.

Slavík, P. (1996). A tight analysis of the greedy algorithm for set cover. Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. Philadelphia, Pennsylvania, United States, ACM.

Wang, N., J. M. Akey, et al. (2002). "Distribution of Recombination Crossovers and the

Origin of Haplotype Blocks: The Interplay of Population History, Recombination, and Mutation." American Journal of Human Genetics, **71**(5): 1227-1234.

Wang, Y., E. Feng, et al. (2007). The Construction of Minimal Set-Covering Model for TagSNP Selection Problem and Heuristic Function Algorithm. Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on.

Weale, M. E., C. Depondt, et al. (2003). "Selection and Evaluation of Tagging SNPs in the Neuronal-Sodium-Channel Gene SCN1A: Implications for Linkage-Disequilibrium Gene Mapping." American Journal of Human Genetics, **73**: 551-565.

Zadeh, L. A. (1963). "Optimality and Nonscalar-Valued Performance Criteria." IEEE Transactions on Automatic Control, **AC-8**(1): 59-60.

Zhang, K., P. Calabrese, et al. (2002a). "Haplotype Block Structure and Its Applications to Association Studies: Power and Study Designs." American Journal of Human Genetics, **71**: 1386-1394.

Zhang, K., M. Deng, et al. (2002). A dynamic programming algorithm for haplotype block partitioning. Proceedings of the National Academy of Sciences. USA. **99:** 7335-7339.

Zhang, K. and L. Jin (2003). "HaploBlockFinder: haplotype block analyses." Bioinformarics, **19**(10): 1300-1301.