國立台東大學資訊管理學系

碩士論文

一致性分割問題之研究

A Study of Consensus Partition Problems

研究生：陳信宏 撰

指導教授：陳彥宏 博士

中華民國九十八年七月

國立台東大學

學位論文考試委員審定書

系所別：資訊管理學系

本班 陳信宏 君

所提之論文 一致性分割問題之研究

業經本委員會通過合於 碩士學位論文 條件

論文學位考試委員會： 張耀中

（學位考試委員會主席）

唐傳義

陳嘉宏

（指導教授）

論文學位考試日期： 98 年 7 月 21 日

國 立 台 東 大 學

博碩士論文授權書

本授權書所授權之論文為本人在 國立臺東大學 資訊管理學系(所)
_____組 九十七 學年度第 二 學期取得 碩 士學位之論文。

論文名稱：_____

本人具有著作財產權之論文全文資料，授權予下列單位：

同意	不同意	單　位
☑	☐	國家圖書館
☑	☐	本人畢業學校圖書館
☑	☐	與本人畢業學校圖書館簽訂合作協議之資料庫業者

得不限地域、時間與次數以微縮、光碟或其他各種數位化方式重製後散布發行或上載網站，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

☐同意 ☐不同意 本人畢業學校圖書館基於學術傳播之目的，在上述範圍內得再授權第三人進行資料重製。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：_____，請將全文資料延後半年再公開。

公開時程

立即公開	一年後公開	二年後公開	三年後公開
	✓		

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，本人同意視同授權。

指導教授姓名：陳義宏　(親筆簽名)

研究生簽名：陳信元　(親筆正楷)

學　　號：9601308　(務必填寫)

日　　期：中華民國 98 年 7 月 25 日

1.本授權書 (得自 http://www.lib.ntu.edu.tw/these/ 下載) 請以黑筆撰寫並影印裝訂於書名頁之次頁。

2.依據 91 學年度第一學期一次校務會議決議:研究生畢業論文「至少需授權學校圖書館數位化，並至遲於三年後上載網路供各界使用及校內瀏覽。」

博碩士論文電子檔案上網授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之論文為授權人在 國立臺東大學 資訊管理學系碩士班 _____ 組 97 學年度第 二 學期取得 碩士 學位之論文。

論文題目： 一致性分割問題之研究
指導教授： 陳彥宏

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館及本人畢業學校圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

- 讀者基非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：陳信宏
簽　名：陳信宏　　中華民國 98 年 07 月 28 日

致謝

首先要感謝我的指導教授　陳彥宏教授這兩年下來的指導，無論是在學問方面、論文的完成上或是待人接物方面，老師都指點我許多。如果說入學前跟畢業後的我有什麼進步的話，我想最大的功勞就是老師。同時也要感謝兩位口試委員，　張耀中教授與　唐傳義教授，百忙之中抽空對我進行口試並且惠賜許多寶貴意見，以及對這篇論文的肯定。

接著感謝研究室的所有人。一年級的時候常常一起在研究室挑燈夜戰的邵永、小洪、宜澤、算厂、佳慧。跟我從大學時代就是同學的海灘、淳瀚。學弟柚子跟學妹千千。雖然到二年下因為我自己的關係很少出現在研究室，不過到了口試當天大家還是很好心的幫我打點。

最後要感謝我的父母親，一直以來他們都在背後默默的關心著我。謹以此論文獻給所有幫助過我的人。

陳信宏 於

資訊管理學系碩士班

2009 年 7 月 28 日

中文摘要

本論文探討兩個一致性分割〔*consensus partition*〕問題，分別為(1) *k partition-distance*問題 (2) *k partition-clustering*問題。這兩個問題可以廣泛的被應用在生物資訊、資料探勘及作業研究上。給定一個集合內有*n*個元素，一個分割(*partition*)是指將此*n*個元素分給多個互斥的群(*cluster*) (每群包含的元素都不同)，最多可分成*n*群。對於相同的元素集合，不同的分割方法將會產生不同的分割，所以在計算不同的分割距離並找出*一致性分割*(*consensus partition*)將會是一項重要的研究。本論文將使用兩個不同的*分割距離*(*partition-distance*)。兩分割P_1及P_2是*一致*(*identical*)定義為在P_1（P_2）中的每個群都會在P_2（P_1）中找到一個相同的群。*第一個分割距離*可定義為移除元素的個數使得移除後兩個分割會是一致的。當有*k>2*個分割時，此分割距離為移除元素的個數使得移除後所有分割會是*一致的*。第一個一致性分割問題，*k partition-distance*問題，定義為移除最少的元素個數，使得*k*個分割會變成*一致的*。如果每個分割有相同個數的群，將兩個分割*對齊*(*alignment*)是指將每個在P_1中的群對應到一個P_2中唯一的群。對於每一對群，我們可以計算這兩群的對稱差集(*symmetric difference*)的元素個數。*第二個分割距離*定義為加總所有對稱差集的元素個數，對於每個在P_1中的群及它在P_2中所對齊的群。當有*k>2*個分割時，此*分割距離*為將所有$\binom{k}{2}$對分割的分割距離加總。第二個一致性分割問題，*k partition-clustering*問題，是尋找最小的第二個分割距離使得*k*個分割會同時對齊。當*k=2*，這兩個問題是一樣的且有時間複雜度為$O(c^3)$的演算法可解決，*c*是2個分割中群的個數加總。當*k>2*，這兩個問題為NP-complete。在此論文中，當*k>2*，我們設計啟發式演算法(heuristic algorithms)來解決這兩個問題，其時間複雜度為$O(k^2n+k\rho \log \rho)$及$O(kn+k\rho \log \rho)$，對於*k partition-clustering*與*k partition-distance*問題並且我們對一些生物上的資料進行模擬測試。

關鍵詞：生物資訊、資料探勘、啟發式演算法、NP-complete、分群、分割、分割距離、一致性分割

Abstract

Given a set of elements *N*, a partition consists of dividing the set of elements into two or more disjoint clusters such that each element belongs to exactly one cluster. A cluster contains a non-empty subset of elements. Different partitioning algorithms for the same application will produce different partitions from the same set of elements. To compute the distance between two or more partitions and find *consensus partition* (also called as *consensus clustering*) are important and interesting problems that arise in many applications such as bioinformatics, data mining and operations research. However, different distance functions between two or more partitions will usually need to design different algorithms. In this thesis, we will study the two consensus partition problems. Given any two partitions P_1 and P_2, the two partitions are *identical* if and only if every cluster in P_1 maps to the same cluster in P_2 (the converse is then forced). *The first partition-distance* of two partitions is the number of elements that need to be deleted from both partitions such that the remaining partitions are identical. If there are more than two partitions, this first partition-distance is defined the number of elements that need to be deleted from all partitions to make them identical. Given a set of *n* elements with *k* partitions, the first consensus partition problem, called as *k partition-distance problem*, is to minimize the first partition-distance. If the number of the clusters of these partitions are equal, an *alignment* of two partitions is each cluster in P_1 matches (*aligns*) to a unique cluster in P_2. We can compute the number of elements of *symmetric difference* for a pair of clusters which are aligned to each other. *The second partition-distance* of two partitions is defined to be the sum of all numbers of elements of symmetric difference for all aligned cluster pairs when both partitions are aligned. If there are more than two partitions, we can align all partitions simultaneously. *The global second partition-distance* is the total second partition-distance summed over all pairs of partitions. Given a set of *n* elements with *k* partitions, the second consensus partition problem, called as the *k partition-clustering problem* is to align all *k* partitions with minimum global second partition-distance among all possible alignments of *k* partitions. If $k=2$, both problems are equivalent and an $O(c^3)$-time algorithm was given, where *c* is the sum of the number of clusters of P_1 and the number of clusters of P_2. If $k>2$, both problems were showed to be NP-complete. In this thesis, we will design heuristic algorithms for both problems applied to some data in bioinformatics research, when $k \geq 2$

Keywords: bioinformatics, data mining, heuristic algorithms, NP-complete, clustering, partition, partition-distance, consensus partition

目　　錄

表目錄

圖目錄

第一章、緒論

在本章節中，我們首先對研究背景與動機做概要的介紹，然後描述本研究所探討的兩種分割問題定義。第二節我們會敘述研究目的，並在第三節說明研究流程。

1.1 研究背景與動機

本研究欲探討的是一致性分割問題(consensus partition)問題，其目的在度量不同分割或分群之間的差異並找尋分割的一致性。分割問題是非常重要的研究且橫跨多個個不同領域，例如:生物資訊，資料探勘及作業研究等。給定一個集合內含有 n 個元素，一個分割是指將此 n 個元素分給多個互斥的群(cluster)(每群包含的元素都不同)。相同的資料在不同的分割演算法執行下將會造成不同的分割。如何計算不同分割之間的距離是令許多學者非常感興趣的重要研究議題。藉由分割距離（*partition-distance*）的計算，我們可以決定(比較)一個分割(方法)的好壞也可以找出這些不同分割的共同的 pattern，稱為*一致性分割*（*consensus partition*）〔或稱作*一致性分群*（*consensus clustering*）〕。本論文將會使用兩種分割距離。第一種分割距離是 Almudevar 與 Field 在 1997 年所提出[1]，用來衡量兩個分割的距離，Gusfield 在 2002 年提出一個一般化的版本，稱為 k partition-distance (k-PD) problem [8]。我們描述此問題如下。給定一個 *n* 個元素(element)的集合 *N*，及兩個分割 P_1 及 P_2，兩個分割 P_1 及 P_2 是*一致(identical)*定義為在 P_1 中的每個群都會在 P_2 中對應到一個相同的群(反之亦然)。Almudevar and Field 定義第一個分割距離函數 d_A：$d_A(P_1,P_2)$-> R^+ ，為移除元素的個數使得移除後兩個分割會變成一致的 [1]。給定 *k* 個分割 $P=\{P_1,P_2,\ldots,P_k\}$及 *n* 個元素的集合，$k\geq 2$，分割距離 $D_A(P)$定義為移除 N 中若干元素使得所有的分割變成一致(identical)。我們正式定義 k-PD 問題如下：

問題：k partition-distance (k-PD) problem
輸入：*k* 個分割 $P=\{P_1,P_2,\ldots,P_k\}$及 *n* 個元素的集合，$k\geq 2$
輸出：使所有分割變成一致(*identical)*
目標：最小化分割距離（$D_A(P)$）

第二種分割距離是 Berman et al.於 2007 年定義的 k partition-clustering (k-PC) problem 問題[5]。我們描述此問題如下：對於兩個分割 P_1 and P_2，如果每一個分割有相同個數的群，每一個分割 P_α有ρ個群$\{C_{\alpha,1},C_{\alpha,2},\ldots,C_{\alpha,\rho}\}$，將兩個分割*對齊*(*alignment*)是指將每個在 P_1 中的群對應(match or *aligned*)到一個 P_2 中唯一的群。對於兩群 $C_{1,i}\in P_1$ 及 $C_{2,j}\in P_2$，對稱差集的元素個數$\nabla(C_{1,i},C_{2,j})=|(C_{1,i}\backslash C_{2,j})\cup(C_{2,j}\backslash C_{1,i})|$。令$\{C'_{1,1}, C'_{1,2},\ldots,C'_{1,\rho}\}$ of P_1 和$\{C'_{2,1}, C'_{2,2},\ldots,C'_{2,\rho}\}$ of P_2 是兩個分割對齊後的情形，Berman et al.定義第二個分割距離函數 d_B：$d_B(P_1,P_2)\text{-> } R^+$，為加總所有對稱差集(symmetric difference)的元素個數，對於每個在 P_1 中的群及它在 P_2 中所對齊的群(i.e., $d_B(P_1,P_2)=\Sigma_{(u=1 \text{ to } \rho)}\nabla(C'_{1u},C'_{2u})$) [5]。給定 *k* 個分割 $P=\{P_1,P_2,\ldots,P_k\}$及 *n* 個元素的集合，$k\geq 2$，每個分割都有相同個數的群(ρ個群)，把所有的分割一起對齊後，*分割距離* $D_B(P)$為將所有$\binom{k}{2}$對分割的分割距離做加總(即 $D_B(P)=\Sigma_{i=1 \text{ to } k}\Sigma_{j=i \text{ to } k}\, d_B(P_i,P_j)$)。我們正式定義 k-PC 問題如下：

問題：k partition-clustering (k-PC) problem
輸入：k 個分割 $P=\{P_1,P_2,\ldots,P_k\}$及 n 個元素的集合，$k\geq 2$，且每次分成相同的群數ρ
輸出：所有 k 個分割會對齊在一起
目標：最小化分割距離（$D_B(P)$）

1.2 研究目的

本研究主要是探討兩個一致性分割問題 k-PD 及 k-PC 問題。這兩個問題可以廣泛的使用在衡量生物資訊上物種的分群。對於 k-PD 問題，當 $k=2$，此問題有多項式時間的演算法可解決，時間複雜度為 $O(c^3)$，*c* 是 2 個分割中群的個數加總，而且 $c=O(n)$ [8]。當 $k>2$，此問題

為 NP-complete [8]。然而 k partition-distance problem 的可近似度集合是否為 Max SNP-complete [13]〔或稱為 APX-complete，沒有 polynomial time approximation scheme (PTAS)，除非 $P=NP$〕，亦或是有 PTAS 的方法至今仍是懸而未知。對於 k-PC 問題，在當 $k=2$，此問題會等同於第一個問題（k-PD problem）[8]，有時間複雜度為 $O(\rho^3)$的演算法可解決，$\rho=O(n)$。當 $k>2$，此問題為 Max SNP-hardness，並有兩倍的近似演算法其時間複雜度為 $O(k^2(n+\rho^3))$ [5]。

對於 k-PD 及 k-PC 問題，我們將設計幾個啟發式演算法(heuristic algorithms)並分析其時間複雜度使其執行速度更有效率。我們的方法是利用分群方法的一些特性來設計這些演算法，其時間複雜度分別為 $O(k^2n+k\rho \log \rho)$ [k-PC 問題]及 $O(kn+k\rho \log \rho)$ [k-PD 問題]，並應用一些生物上的資料來進行模擬(*simulation*)，藉此結果說明我們的方法的優劣。

1.3 研究架構

本論文的研究架構如下，首先將會逐一介紹分群(clustering)在各個領域中應用的情況；對相關工作做一個概要的整理。接著會介紹與本研究關係極為密切的 k-PD 問題與 k-PC 問題。最後將會概述 Berman 等人提出的 k-PC 問題的近似演算法。由於在 $k>2$ 的時候，k-PC 問題是 NP-complete，該演算法能保證所找到的解將會是(2-k/2)倍近似。第三章，首先我們將會提出一個可能的方式，用來代表每個群集的相似程度。然後以該數值為基礎建立一個演算法，先透過在 $k=2$ 的情況下與 Gusfield 演算法進行比較，確定該數值可以在此一情況下運作。最後在與 Berman 演算法進行比較，確定在 $k>2$ 的時候，我們的演算法將如同 Berman et al.的演算法一般找出(2-k/2)倍近似的解。在本論文中的第三章，我們將會展示出我們設計用來表示群集相似程度的數值計算方法，以及一個以該數值為基礎的演算法來解決 k-PD 及 k-PC 問題，並藉此分析演算法的時間複雜度。第四章我們會展示整個實驗數據的模擬結果，藉此觀察我們的方法的執行效率與正確性。最後，我們將在第五章提出結論。

第二章、文獻回顧

本章首先對分群演算法的相關工作做一個概略整理。在第 2 小節、第 3 小節分別對 k-PD 以及 k-PC 問題的相關文獻進行探討。

2.1 分群演算法

分群被廣泛應用在各學科的研究之中，是一個十分重要的研究問題。簡單來說，分群是指將收集好的資料(物件、元素等)，透過分群演算法分成多個群（group）。相同群中的資料擁有高的相似性，不同群的資料彼此之間的差異性會比相同群之間的資料來的大。常見的分群方式以有兩種，一種是 partition-based 的分群[9, 15]，一種是階層式的分群[9, 15]。給定一集合 N 內包含 n 個元素，分割(partition)指把這個集合中的資料分成兩個或更多個互斥的群(cluster) (N 的非空子集)，也就是每個元素必須屬於某一個群[9, 15]。階層式的分群方式則是將給定的集合建構成一個巢狀結構的群集。本研究建構在 partition-based 的分群下。

在生物資訊，資料探勘與作業研究的領域上，有相當多的分群演算法已被提出。Alumdevar 與 Field 提出一個分群演算法用來建構魚類的家族血緣關係，透過 DNA markers 進行分群[1]。Konovalov，Manning 與 Henshaw 實做一個 java 程式的分群方法，透過評估所有分群結果的可能性重新建構物種的演化關係[10]，Beyer 與 May 利用圖論方法將一組資料中的個體依照同親關係分組，他們使用 single-locus co-dominant markers 進行分組[6]。Bagirov 與 Mardaneh 設計一個改良的全域 k-mean 演算法應用在肺癌、白血病的基因表現資料上[2]。Ben-Dor，Yakhin 提出一個多條件的基因表現樣式(gane expression patterns)的分群問題，並且提出一個階層式分群演算法來解決這個問題[3]。Bulter et. al 使用 Almudevar 與 Field 的演算法來比較四種不同的分群演算法，這些分群演算法是用 DNA marker 資料庫重新建立同親族譜（full sib pedigrees）[4]。Vakharik 與 Mahajan 建構一個新的分群演算法，它可以同時對資料以及屬性集合進行分群，此分群可應用在零售業與製造業[16]。Saglem et al.提出一個混合整數規劃模型將顧客分成數個類別並應用在顧客關係管理(CRM)[14]。

以下分別展示在不同領域，一些歷年來前人做過的相關研究工作:

作者	發表時間	領域	內容概述
Alumdevar，Field	1999 年	生物資訊	提出了一個指數時間的演算法，去找出分割距離。在他們的研究中，採用的是有關漁業養殖的資料，並且將物種資料依照血親關係分割成多個群 [1] 。
Bagirov，Mardaneh	2006 年	生物資訊	提出一個全域的 k-mean 分群演算法，應用在基因表現資料的分群上，他們採用的實際資料是有關肺癌與白血病的資料庫 [2] 。
Ben-Dor，Yakhin	1999 年	生物資訊	描述一個對多條件(multi-condition)基因表現樣式的分群問題，並提出一個新的階層式分群演算法解決分群結構問題並且進行測試 [3] 。
Beyer，May	2003 年	生物資訊	運用圖形理論(graph-theoretic)方法去把物種親緣家譜，它使用生物資訊領域的 single-locus co-dominant markers [6] 。
Konovalov，Manning，Henshaw	2004 年	生物資訊	利用 java 程式實做一個新的方法，評估了所有可能的分割情況後，重新建構物種演化關係 [10] 。
Olafsson，Li，Wu	2008 年	作業研究與資料探勘	提供一個近期作業研究領域應用分群演算法的相關文獻整理 [12] 。
Vakharik，Mahajan	2000 年	作業研究	建構一個新的分群演算法，它可以同時對資料以及屬性集合進行分群，應用在零售業與製造業 [16] 。
Saglem et al.	2006 年	作業研究	提出一個混合整數規劃模型將顧客分成數個類別，應用在顧客關係管理(CRM) [14] 。

表 1. 相關研究整理

2.2 k-PD 問題

不同的分群演算法(partition-based)如果應用在同一個給定的資料集合，可以預期的會產生許多不同的分群結果。要如何從眾多的分群結果中找出最好的，是一個重要而且有趣的問題。為了解決這個問題，我們需要分割距離(partition-distance)來對兩個或更多個分割進行衡量。

Almudevar 與 Field 提出定義了一種分割距離是在 n 個元素的集合中，為了使兩個分割成為一致最少需要移除的元素個數 [1]。給定一個集合含 n 個元素，當我們對該集合進行兩次分群，得到兩種不一樣的分群結果 P_1 與 P_2，將兩分割對齊之後，計算需要從集合移除掉最少個元素，這兩個分割會完全相等，而最少需要被移除的元素個數就是其分割距離。針對此問題，Almudevar 與 Field 提出了一個指數時間的演算法去找出兩個分割的分割距離。在他們的研究中，採用的是有關漁業養殖的資料，並且將物種資料依照血親關係分割成多個群。Bulter et al. 應用 Almudevar 與 Field 的演算法透過一些相關的參數，去比較四種分割演算法，進而從 DNA marker 資料中重新建構同親(同個父母所生)家譜（full sib pedigrees）[4]。Gusfield 提出一個 $O(c^3)$的演算法，轉換分割距離問題到最大分配（maximum weighted assignment）的問題[8]，這裡的 c 是指 P_1 所包含的群集數目以及 P_2 所包含的群集數目的加總。Konovalov，Litow 與 Bajema 也獨立設計出了一個同樣是 $O(c^3)$時間的演算法（轉換到最小分配（minimum weighted assignment）的問題）[11]。Gusfield 同時將兩個分割的距離一般化成 k 個分割距離問題，稱為 k partition-distance (簡稱 k-PD 問題) [8]。$k>2$ 的時候，Gusfield 透過轉換 3 維配對問題(3-dimensional matching)到此問題來證明 k-PD 問題是 NP-complete [8]。我們用一個例子來詳述此問題。

Example:給定一集合 N={1, 2, 3, 4, 5, 6, 7, 8, 9 }，對 N 進行兩種分群演算法候得到一個分割的集合 P={P_1,P_2}。P_1 有 3 個 clusters $C_{1,1}$、$C_{1,2}$ 及 $C_{1,3}$，P_2 也有 3 個 clusters $C_{2,1}$、$C_{2,2}$ 及 $C_{2,3}$。表 2.中，我們移除 {4, 5, 6, 7, 8, 9}使得 P_1 和 P_2 變成一致，所以 $d_A(P_1,P_2)$ 是 6。表 3.顯現一個最佳解對於 2-PD 問題，我們移除元素 {3, 4, 5}使得 P_1 和 P_2 變成一致，因此 $d_A(P_1,P_2)$ 是

3。表 4.是另一個例子用來說明 3-PD 問題。我們有一個分割集合 $P=\{P_1,P_2,P_3\}$及一個元素集合 $N=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$。我們移除$\{4, 5, 8, 9, 0\}$使得 P_1、P_2 及 P_3 變成一致，所以 $D_A(P)$ 是 5，此分割距離同時也是一個最佳解。

P_1	P_2	$d_A(P_1,P_2)$
$C_{1,1}=\{1,2\}$	$C_{2,1}=\{1,2,4,5\}$	移除{4,5}
$C_{1,2}=\{4,5,6,7\}$	$C_{2,2}=\{8,9\}$	移除{4,5,6,7,8,9}
$C_{1,3}=\{3,8,9\}$	$C_{2,3}=\{3,6,7\}$	移除{6,7,8,9}
		$d_A(P_1,P_2)=6$

表 2. 一個移除的方式對於 2-PD 問題，分割距離為 6。

P_1	P_2	$d_A(P_1,P_2)$
$C_{1,1}=\{1,2\}$	$C_{2,1}=\{1,2,4,5\}$	移除{4,5}
$C_{1,2}=\{4,5,6,7\}$	$C_{2,3}=\{3,6,7\}$	移除{3,4,5}
$C_{1,3}=\{3,8,9\}$	$C_{2,2}=\{8,9\}$	移除{3}
		$d_A(P_1,P_2)=3$

表 3. 最佳的移除的方式對於 2-PD 問題，分割距離為 3。

P_1	P_2	P_3	$D_A(p)$
$C_{1,1}$={1,2,8,9,0}	$C_{2,2}$={1,2,4,5}	$C_{3,1}$={1,2}	移除 {4,5,8,9,0}
$C_{1,2}$={4,5}	$C_{2,1}$={8,9,0}	$C_{3,2}$={4,8}	移除 {4,5,8,9,0}
$C_{1,3}$={3,6,7}	$C_{2,3}$={3,6,7}	$C_{3,3}$={3,5,6,7,9,0}	移除{9,0}
			$D_A(P)$=5 移除 {4,5,8,9,0}

表 4. 最佳的移除的方式對於 3-PD 問題，分割距離為 5。

2.3 k-PC 問題

Berman 等人提出 k partition-clustering problem (簡稱 k-PC 問題)是尋找最小分割距離，使得所有 k 個分割會*對齊*在一起。這裡的分割距離是將所有分割中的所有的群對齊後計算其對稱差集的元素個數的加總 [5]。針對 $k>2$，他們提出了一個提出了一個(2-k/2)倍近似演算法。並且說明當 $k=2$，2-PD 問題將等價於 2-PC 問題。他們也提出對於 k-PC 問題在 $k>2$ 的時候是此問題為 Max SNP-hard。對於最小化問題（minimization problem），若一個演算法是 r 倍近似，代表它所產出的解永遠會小於等於最佳解的 r 倍 [7]。這裡的 k 為分割的數目。Berman 等人的近似演算法應用 Gusfield 的 2-PD 問題的正確演算法，首先對 k 個分割進行 k^2 次運算，透過 assignment 問題的解找出兩兩分割間的最佳對齊序列。因為 k-PC 問題要求每個分割的群集數相同，將兩個分割的群集對齊之後，計算兩兩對齊的群集的對稱差集的元素個數，然後把他們加總起來[5]。我們列出他們的演算法如下。

- 對於每個 $1\le i\le k$, 針對每個 P_i 分別去找他們跟每個 P_j，$j\neq i$，的最佳對齊方式(透過 2-PD 問題的演算法)。

- 找出這些解中最好(小)的一個。

對的 k-PC 問題，他們證明了此方法有(2-2/k)倍近似，證明概述如下：

變數與目標函數定義：

- 令有 k 個分割 $S_1,...,S_k$。
- 每個分割包含ρ個群集。
- 令 $\sigma = (\sigma_1, \ldots, \sigma_k)$為一組排序可使目標函數最佳化

  $OPT = \Sigma_{i=1 \text{ to } q} \Sigma_{1 \le j < r \le k} \Delta(S_{j,\sigma^{\square}(i)}, S_{r,\sigma^{\square}(i)})$。
- 對任何 k 分割 $\sigma = (\sigma_1,\sigma_2, \ldots, \sigma_k)$為$\{1, 2, \ldots, \rho\}$的排列，

  $\Delta\sigma (P_j, P_r) = \Sigma_{i=1 \text{ to } q} \Delta(S_{j,\sigma j (i)}, S_{r,\sigma r (i)})$。

證明：演算法為(2-k/2)倍近似

假設k>2。透過演算法求出的解Δ明顯滿足三角不等式，意指對任何三個集合X，Y，Z，

$\Delta (X, Z) \le \Delta(X, Y) + \Delta(Y, Z)$。

若最佳解為$\sigma_r = (\sigma_{r,1},\sigma_{r,2}, \ldots, \sigma_{r,k})$

對任何的 i 與 j ， $\Delta\sigma_i (P_i,P_j) = \Delta\sigma_j (P_i,P_j)$。

對任何的 i 與 j ， $\Delta\sigma_i (P_i,P_j) \le \Delta\sigma(P_i,P_j)$且

$$\Sigma_{1 \le i < j \le k} \Delta\sigma_i (P_i,P_j) \le \Sigma_{1 \le i < j \le k} \Delta\sigma^{\square}(P_i,P_j) = OPT \qquad (1)$$

對任何的i，j，r ，

$$\Delta\sigma_i (P_j,P_r) = \Sigma_{i=1 \text{ to } q} \Delta(S_{j,\sigma i},S_{r,\sigma i})$$

$$\le \Sigma_{i=1 \text{ to } q} \Delta(S_{i,\sigma i},S_{j,\sigma i}) + \Delta(S_{i,\sigma i},S_{r,\sigma i})$$

$$= \Delta\sigma_i (P_i,P_j) + \Delta\sigma_i (P_i,P_r) \qquad (2)$$

對任何的 i，

$$f(\sigma_i) = \Sigma_{s \ne i} \Delta\sigma_i (P_i,P_s) + \Sigma_{j \ne i} \Sigma_{r \ne i,\, r > j} \Delta\sigma_i (P_j,P_r)$$

$$\le \Sigma_{s \ne i} \Delta\sigma_i (P_i,P_s) + \Sigma_{j \ne i} \Sigma_{r \ne i,\, r > j} [\Delta\sigma_i (P_i,P_j) + \Delta\sigma_i (P_i,P_r)] \quad \text{(透過不等式(2))}$$

$$= \Sigma_{s \ne i} \Delta\sigma_i (P_i,P_s) + (k-2) \cdot \Sigma_{j \ne i} \Delta\sigma_i (P_i,P_j)$$

$$= (k-1) \cdot \Sigma_{j \ne i} \Delta\sigma_i (P_i,P_j)$$

因此，

$$\Sigma_{i=1 \text{ to } k} f(\sigma_i) \le \Sigma_{i=1 \text{ to } k}[(k-1) \cdot \Sigma_{j \ne i} \Delta\sigma_i (P_i,P_j)]$$

$$= (k-1) \cdot \Sigma_{i=1 \text{ to } k} \Sigma_{j \ne i} \sigma_i (P_i,P_j)$$

$$= 2 \cdot (k-1) \cdot \Sigma_{1 \le i < j \le k} \sigma_i (P_i,P_j)$$

〔因為 $\Delta\sigma_i (P_i,P_j) = \Delta\sigma_j (P_j,P_i)$〕

$$\le (2k-2) \cdot OPT \qquad \text{(透過不等式(1))}$$

且 $f(\sigma_r) \le (1/k) \cdot f(\sigma_r) \le (2-/2) \cdot OPT$ #

演算法時間則為複雜度為 $O(k^2(n+\rho^3))$。 針對 k-PC 問題，我們也用一個例子來詳述此問題。

Example:給定一集合 N={1, 2, 3, 4, 5, 6, 7, 8, 9 }，對 N 進行兩種分群演算法候得到一個分割的集合 P={P_1,P_2}。P_1 有 3 個 clusters $C_{1,1}$、$C_{1,2}$ 及 $C_{1,3}$，P_2 也有 3 個 clusters $C_{2,1}$、$C_{2,2}$ 及 $C_{2,3}$。表 5.中，我們將 $C_{1,1}$ 對齊 $C_{2,1}$，$C_{1,2}$ 對齊 $C_{2,2}$，及 $C_{1,3}$ 對齊 $C_{2,3}$，所以 $d_B(P_1,P_2)$是 12。表 6.顯現一個最佳解對於 2-PC 問題，我們將 $C_{1,1}$ 對齊 $C_{2,1}$，$C_{1,2}$ 對齊 $C_{2,3}$，及 $C_{1,3}$ 對 $C_{2,2}$，因此 $d_B(P_1,P_2)$是 6。表 7.是另一個例子用來說明 3-PC 問題。我們有一個分割集合 P={P_1,P_2,P_3}及一個元素集合 *N*={1, 2, 3, 4, 5, 6, 7, 8, 9, 0}。我們對齊{$C_{1,1}$, $C_{2,1}$, $C_{3,2}$}，{$C_{1,2}$, $C_{2,2}$, $C_{3,1}$}，及 {$C_{1,3}$, $C_{2,3}$, $C_{3,3}$}，因此 $D_B(P)$是 24，此距離同時也是一個最佳解。從這些範例中，我們可以發現 2-PC 問題的距離是 2-PD 問題的 2 倍。這不難證明當 *k=2* 時這兩個問題是等價的。

P_1	P_2	$d_B(P_1,P_2)$
$C_{1,1}$={1,2}	$C_{2,1}$={1,2,4,5}	$\nabla(C_{1,1},C_{2,1})$=2
$C_{1,2}$={4,5,6,7}	$C_{2,2}$={8,9}	$\nabla(C_{1,1},C_{2,1})$=6
$C_{1,3}$={3,8,9}	$C_{2,3}$={3,6,7}	$\nabla(C_{1,1},C_{2,1})$=4
		$d_B(P_1,P_2)$=12

表 5. 一個對齊的方式對於 2-PC 問題，分割距離為 12

P_1	P_2	$d_B(P_1,P_2)$
$C_{1,1}=\{1,2\}$	$C_{2,1}=\{1,2,4,5\}$	$\nabla(C_{1,1},C_{2,1})=2$
$C_{1,2}=\{4,5,6,7\}$	$C_{2,3}=\{3,6,7\}$	$\nabla(C_{1,1},C_{2,1})=3$
$C_{1,3}=\{3,8,9\}$	$C_{2,2}=\{8,9\}$	$\nabla(C_{1,1},C_{2,1})=1$
		$d_B(P_1,P_2)=6$

表 6. 最佳對齊的方式對於 2-PC 問題，分割距離為 6。

P_1	P_2	P_3	$D_B(P)$
$C_{1,1}=\{1,2,8,9,0\}$	$C_{2,1}=\{8,9,0\}$	$C_{3,2}=\{4,8\}$	
$C_{1,2}=\{4,5\}$	$C_{2,2}=\{1,2,4,5\}$	$C_{3,1}=\{1,2\}$	
$C_{1,3}=\{3,6,7\}$	$C_{2,3}=\{3,6,7\}$	$C_{3,3}=\{3,5,6,7,9,0\}$	
			$D_B(P)=$ $d_B(P_1,P_2)+$ $d_B(P_1,P_3)+$ $d_B(P_2,P_3)=24$

表 7. 最佳對齊的方式對於 3-PC 問題，分割距離為 24。

第三章、研究方法

本章中，第一小節說明我們預計解決此兩個問題的方法概述。我們提出的演算法在第二小節說明，第三節描述此演算法的時間複雜度。

3.1 方法概述

Berman 等人 [5]的演算法中，如果我們先將所有的分割的群對齊來搜尋，之後再把兩個對齊的群中的對稱差集去除，則所有的分割將會變成一致。因此我們可以藉由此一概念，我們只需要討論k-PC問題即可，當得到k-PC問題的解時，我們可以藉由上述的方法轉換成k-PD問題的解。然而因為k-PC問題要求所有的分割的群集數都要相同，而k-PD則無此要求，因此在不失一般性下，我們假設所有的分割的群集數都要相同，若不相同，我們可以藉由補一些空集合的群到每個分割中。我們的想法是：如果我們把每個分割中的群集都給定一個數值，此一數值可以用來代表該群，因此分割中群，如果其數值越接近，將其對齊後的對稱差集應該會越少。因此對於每個分割，我們分別把它的群集代表的數值排序，然後依照排序結果，將所有分割對齊。這樣的方法我們會是一個多項式的時間的演算法，且執行速度將會比 $O(k^2(n+\rho^3))$ time (i.e., Berman 等人提出的近似演算法的時間複雜度 [5])快很多(第三節將會說明)。

本研究的方法可以歸納如下

1. 找出一個足以代表群集特性的代表數值，數值越相近的群集，其群內包含的元素越相似。
2. 設計一個能在多個分割的情形($k>2$)，透過數值排序，找出一種對齊方式，之後計算其對稱差集，找到k-PC問題的分割距離。
3. 設計一個能在多個分割的問題中(k>2)，透過數值排序，找出一種對齊方式，之後找出其對稱差集的元素，將這些元素去除，找到k-PD問題的分割距離。

3.2 演算法

演算法如下:

1. 給定集合包含 n 個元素，對 n 個元素做編號從 1 到 n。
2. 對每個分割中的ρ個群集各別計算一個特徵值 s。

   （特徵值 s＝群集中元素編號的平均值+群集中元素的間距總和。）
3. 分別將分割中的群集依照 s 的大小進行排序。
4. 所有的分割依照群集排序後的結果對齊。
5. 計算對齊後的兩群的對稱差集，將其加總得到 k-PC 問題的分割距離
6. 找出對齊後的兩群的對稱差集，將其刪除，使得所有分割一致，刪除的個數即為 k-PD 問題的分割距離。

我們提出的演算法屬於啟發式演算法。啟發式演算法的設計常常是透過模仿自然界中的現象或規則，眾所皆知的幾個例子為：遺傳基因演算法、模擬退火法、螞蟻演算法等等...。這類型演算法常常可以在合理的時間內求出不錯的解，然而無法確定該演算法不會找到更壞的解，或是每次都能在合理時間內求解。

特徵值 s 的設計構想來自於利用一個數值來表達集合的特色。透過直接的觀察我們可以發現兩個集合如果相同，則他們的所包含的元素個數與元素編號總和必定會相同。但是這兩個因素都相同的集合不一定就相同，例如：集合 A={1,5,9}，集合 B={2,4,9}，兩集合的元素個數都是 3，且元素編號總和都是 15，但兩者明顯不同。從這個例子我們很直覺的觀察到集合 A 與集合 B 兩兩元素編號的間距不同，集合 A 的元素間距為(4,4)，集合 B 元素間距(2,5)，所以我們將同集合的元素編號間距以加總的方式整合，也就是集合 A 的元素編號間距和為 8，集合 B 元素編號間距和為 7。我們假設以上述三個因素元素個數、元素編號總和、元素編號間距和在某種程度上足以表達群集的特性，這部份將透過實驗來驗證。排序目的在於讓特徵值相近的群集能夠對齊，只要兩個分割中的群集是用相同的方法排序，無論由大到小或由小到大，對結果都沒有影響。

底下舉例說明我們的演算法。給定一集合 N={1, 2, 3, 4, 5, 6, 7, 8, 9 }，對 N 進行兩種分群演算法候得到一個分割的集合 $P=\{P_1,P_2\}$。P_1 有 3 個 clusters $C_{1,1}$、$C_{1,2}$ 及 $C_{1,3}$，P_2 也有 3 個 clusters $C_{2,1}$、$C_{2,2}$ 及 $C_{2,3}$。

1. 編碼 9 個元素 sourceset={1,2,3,4,5,6,7,8,9}。
2. 計算分割中每個群集ρ的 s 值 (s=群集中元素編號的平均值+群集中元素的間距總合。) (表 8)。
3. 依照 s 排序(表 9)。
4. 所有的分割依照群集排序後的結果對齊(表 9)。
5. 計算其對稱差集的加總 $D_B(P)$。
6. 計算被刪除的元素個數的加總 $D_A(P)$。

P_1	s	P_2	s
$C_{1,1}=\{1,2\}$	$s_{1,1}=3/2+1$ =2.5	$C_{2,1}=\{1,2,4,5\}$	$s_{2,1}=12/4+4=7$
$C_{1,2}=\{4,5,6,7\}$	$s_{1,2}=22/4+3$ =8.5	$C_{2,2}=\{8,9\}$	$s_{2,2}=17/2+1$ =9.5
$C_{1,3}=\{3,8,9\}$	$s_{1,3}=20/3+6$ =12.67	$C_{2,3}=\{3,6,7\}$	$s_{2,3}=16/3+4$ =9.33

表 8.一個範例針對我們的演算法（計算每個群的特徵值）。

P_1	s	P_2	s
$C_{1,1}$={1,2}	$s_{1,1}$=2.5	$C_{2,1}$={1,2,4,5}	$s_{2,1}$=7
$C_{1,2}$={4,5,6,7}	$s_{1,2}$=8.5	$C_{2,3}$={3,6,7}	$s_{2,3}$=9.33
$C_{1,3}$={3,8,9}	$s_{1,3}$=12.67	$C_{2,2}$={8,9}	$s_{2,2}$=9.5

表 9.一個範例針對我們的演算法（每個分割依照特徵值將其內部的群排序）。

P_1	P_2
$C_{1,1}$={1,2}	$C_{2,1}$={1,2,4,5}
$C_{1,2}$={4,5,6,7}	$C_{2,3}$={3,6,7}
$C_{1,3}$={3,8,9}	$C_{2,2}$={8,9}

表 10.一個範例針對我們的演算法（依照排序將每個分割對齊）。

透過上述例子我們可以看出這樣對齊之後，取得的 $D_B(P)$是 6。刪除{3,4,5}，$D_A(P)$將會是 3，在此例中，兩個分割距離分別為 2-PD 及 2-PC 問題的最佳解。

3.3 演算法的時間複雜度

因為 k-PC 與 k-PD 問題被證明為 NP-complete，所以目前無法在多項式的時間複雜度下找到此兩問題的最佳解（optimal solution），當 k>2 時。然而我們設計的啟發式演算法，對於 k-PC 問題，可以在 $O(k^2n+k\rho \log \rho))$ time 找到一組合理的解（feasible solution），對於 k-PD 問題，可以在 $O(k(\rho \log \rho+n))$ time 找到一組合理的解（feasible solution）。雖然找到的解不是最佳解，然而以時間複雜度來看，我們的方法比 Berman 等人提出的近似演算法的時間複

雜度快很多，且 $k=2$ 時也比求出最佳解演算法的時間快的多。底下我們分析第二節的演算法的時間複雜度。

演算法如下:

1. 給定集合包含 n 個元素，對 n 個元素做編號從 1 到 n。
2. 對每個分割中的ρ個群集各別計算一個特徵值 s。

   （特徵值 s＝群集中元素編號的平均值+群集中元素的間距總和。）
3. 分別將分割中的群集依照 s 的大小進行排序。
4. 所有的分割依照群集排序後的結果對齊。
5. 計算對齊後的兩群的對稱差集，將其加總得到 k-PC 問題的分割距離
6. 找出對齊後的兩群的對稱差集，將其刪除，使得所有分割一致，刪除的個數即為 k-PD 問題的分割距離。

在此演算法中，第一步驟，我們花了 $O(n)$ time 來編排元素。在第二步驟，對於一個分割，要算出每個分割內的群集的特徵值要花 $O(n)$ time，因為我們有 k 個分割，所以第二步驟將花費 $O(kn)$ time。第三步驟的排序每個分割內所有群集的特徵值需要 $O(\rho \log \rho)$ (因為最佳的排序演算法的時間複雜度為 $O(\rho \log \rho)$ time)，所以第三步驟將花費 $O(k\rho \log \rho)$ time。第四步驟的對齊只要花 $O(k\rho)$ time 即可。第五步驟為要求出 k-PC 問題的一組解。對於兩個分割，我們要算出他們對齊後的群集的對稱差集需花費 $O(n)$ time，對於任兩個分割的組合，我們均需分別求其對稱差集，因此所需的總時間為 $O(k^2n)$ time；因此針對 k-PC 問題，我們的方法將可在 $O(k^2n+k\rho \log \rho)$ time 完成。第六步驟為要求出 k-PD 問題的一組解。對於兩個分割，我們要算出他們對齊後的群集的交集需花費 $O(n)$ time，刪掉對稱差集的時間也需 $O(n)$ time。對於兩個分割，當我們可以先對第一個與第二個分割使其一致性後，再使第一個跟第三個分割的達到一致性（因為第二個分割已經和第一個分割產生一致性了，兩分割已經一模一樣的）。所以，我們只需要再對第一個跟第三個分割使其一致性。如此，當做完第一個跟第 k 個分割的

一致性後，所有的分割將會全部達到一致，所需的總時間將降為 $O(kn)$ time；因此針對 k-PD 問題，我們的方法將可在 $O(kn+k\rho \log \rho)$ time 完成。

第四章 實測驗證

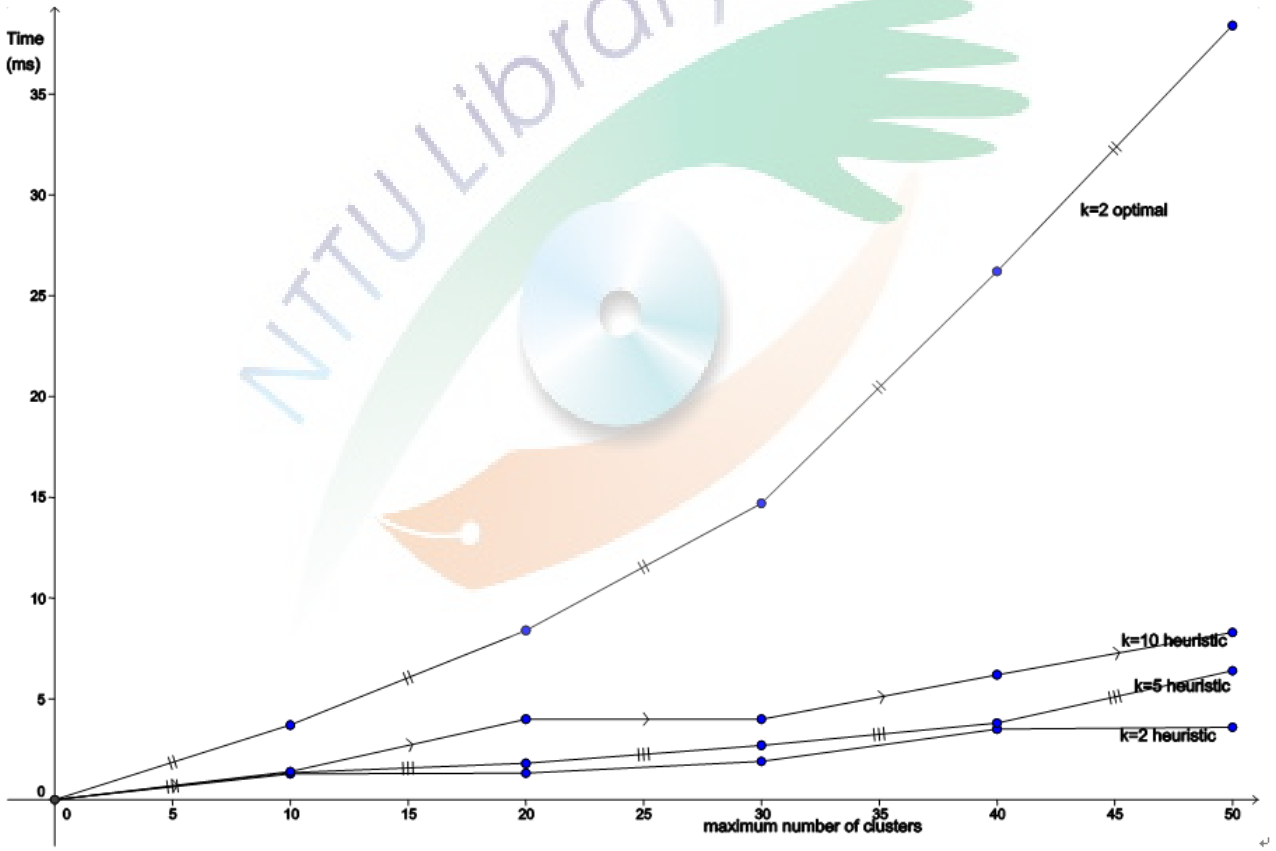
本章中，我們將針對我們設計的演算法與 2-PC (2-PD)問題求得最佳解的演算法來進行實際的程式模擬，因為 k-PC (k-PD)問題在 $k>2$ 時為 NP-complete，且尚無正確演算法（exact algorithm）被提出，所以我們將用 2-PC 問題的正確演算法[8]來與我們的演算法進行比較。因為 2-PC 問題是被轉換到最大分配（maximum weighted assignment）的問題，此問題可透過匈牙利演算法（Hungarian algorithm）解決，所以針對 2-PC 問題，我們實作匈牙利演算法的程式。此演算法的時間複雜度為 $O(\rho^3+n)$ time。雖然實作結果顯示，我們的方法無法達到最佳解，然而我們的方法其時間卻比最佳解的演算法快的多。這些程式都被執行在相同的機器上。工作站型號：Sun Microsystems SPARC Station-20 Model 71，CPU 型態：TI Model 71 SuperSPARC-II SPARCmodule 75 MHz，作業系統：Solaris 7。

現在，我們令 $D(\rho,N_{max})$ 為一個分割有 ρ 個群集。對於每個群集，都有其元素個數，所有的群集中最多能有 N_{max} 元素個數。我們使用的 data 是從 But1er 等人[4]及 Konovalov 等人[11]的論文中提出的。在第一節中，第一個分割我們使用了 $D(\rho,1)$，其餘的分割則是亂數產生，藉此找出這些分割的一致性分割。在第二節中，第一個分割我們使用了 $D(\rho,10)$，其餘的分割則是亂數產生，藉此找出這些分割的一致性分割。第三節中，為了讓生物學家逕行尋找一致性分割，我們也提供了一個網站來供生物學家使用我們的程式，此網站的程式是由 PHP 寫成，網址為 http://210.240.178.54/kpc.html。

4.1 實測 D(ρ,1)案例

本節中，第一個分割是取自不相關的物種及不充分的基因資訊 (small number of loci and/or alleles) [4,11]。因此，我們建構第一個模擬的分割 $D(\rho,1)$，其餘的模擬資料是亂數產生。在亂數的分割中，針對每個分割將產生 ρ 個元素，每個元素亂數的放到一個群集內。在此案

例中，對於 2-PD 與 2-PC 問題在最糟的情形下(worse case)，Gusfield 的演算法（找出最佳解的演算法）執行時間為 $O(n^3)$ time [8]，我們的方法執行時間為 $O(n \log n)$ time。圖一中列出了我們的演算法與 Gusfield 演算法之時間的比較結果。舉例如下，Butler 等人 [4]重建了五十個不相關物種之分群，所以第一個分割為 D(50,1)。其餘的分割為亂數產生。針對這個案例，我們執行了 100 次亂數的資料與 D(50,1)求一致性分割並進行速度的測試。在此結果下，我們的演算法平均需花 3.4 ms 的時間，而 Gusfield 的演算法（產生最佳解）平均需花 38.4 ms，對於 $k=2$。我們也對 $k=5$ 與 $k=10$ 進行模擬，其時間分別為 6.4ms 與 8.3ms。很清楚的我們的方法確實比 Gusfield 的方法在執行的時間上有效率的多。



圖一：k-PD 問題的時間度量的模擬，針對 D(ρ,1)資料集。第一個分割為 D(ρ,1)，其餘的分割為亂數產生，每個點的值是執行 100 組資料後的平均值，標準差為 0.0000315。

為了說明我們的演算法所產生的解與最佳解之間並不會相距太遠，表 11.列出了我們的啟發式演算法所產生出的解與最佳解（$k=2$）之間的 ratio（我們的解除以最佳解）。實驗結果顯示這個演算法與最佳解間的距離(或對稱差集的個數)差距大約在 2~3 倍的比率之間，雖然我們未能以數學證明我們的解距離最佳解的保證，然而實驗數據顯示我們的解仍不失為一個好的結果。

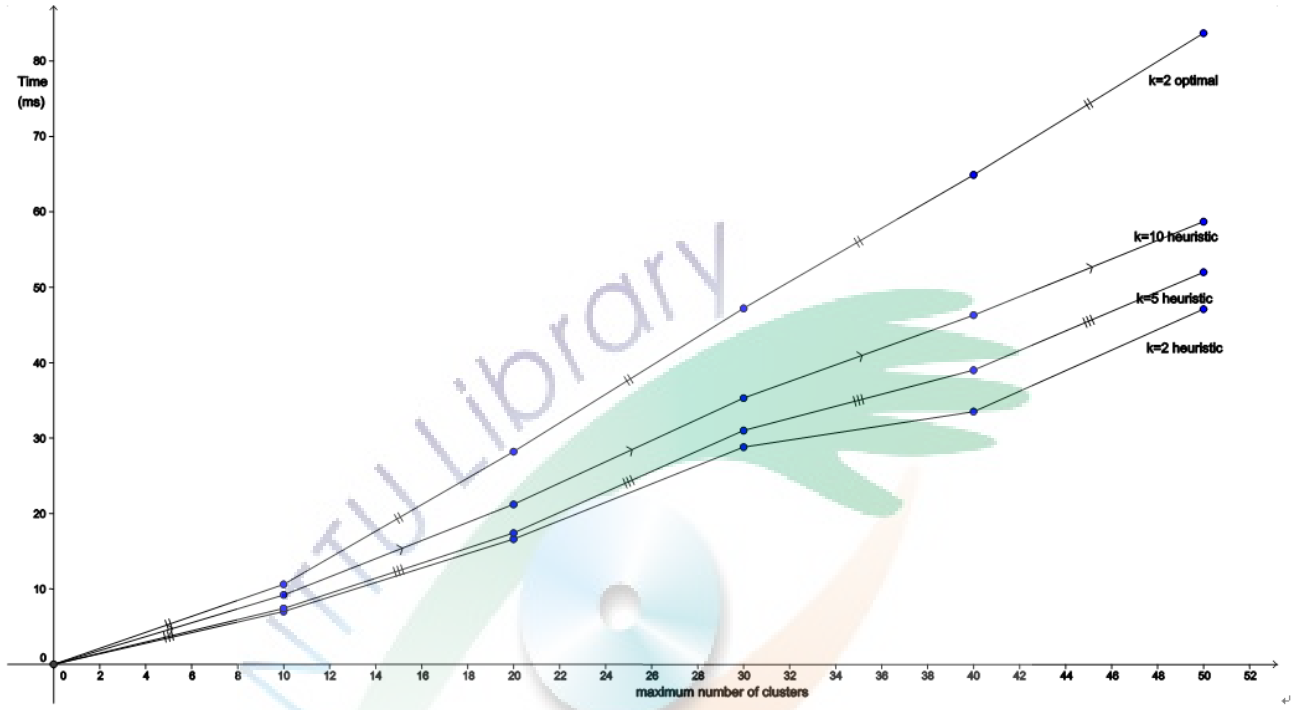
ρ	ratio(our heuristic solution/optimal solution)
10	2.648
20	2.676
30	2.541
40	2.849
50	2.696

表 11. 2-PD 問題的模擬結果(找出最佳解與我們的解的差異)，針對 D(ρ,1)資料集。ratio 此為我們的解除以最佳解。第一個分割為 D(ρ,1)，其餘的分割為亂數產生，每個數據的值是執行 100 組資料後的平均值，標準差為 0.17304。

4.2 實測 D(ρ,10)案例

本節中，第一個分割取自此 Butler 等人[4]及 Konovalov 等人[11]的論文中可能的產出的分群結果。此結果為每個群集包含十個物種(元素)。因此，我們可以建構第一個模擬的分割為 D(ρ,10)，其餘的模擬資料是亂數產生。對於每個分割是產生ρ個元素，每個元素亂數的放到一個群集內，每個群集最多包含十個元素。圖二中列出了我們的演算法與 Gusfield 演算法之時間的比較結果。舉例如下，Konovalov 等人[11]重建了 50*10 個物種之分割，這個分割包含 50 個家族(群集)，每個群集內最多十個物種。所以第一個分割為 D(50,10)。針對這個案例，

我們執行了100次亂數產生的分割與D(50,10) 求一致性分割並進行速度的測試。在此結果下，我們的演算法平均需花47 ms 的時間，而 Gusfield 的演算法(產生最佳解)平均需花 83.7 ms，對於 *k=2*。我們也對 *k=5* 與 *k=10* 進行模擬，其時間分別為 52.1ms 與 58.8ms。很清楚的我們的方法確實比 Gusfield 的方法在執行上速度上快很多。



圖二：k-PD 問題的時間度量的模擬，針對 D(ρ,10)資料集。第一個分割為 D(ρ,10)，其餘的分割為亂數產生，每一個數據的值是執行 100 組資料後的平均值，標準差為 0.0000974。

同樣的，為了說明我們的演算法所產生的解與最佳解之間並不會相距太遠，表 12.列出了我們的啟發式演算法所產生出的解與最佳解的 ratio（我們的解除以最佳解）。實驗結果顯示這個演算法與最佳解間的距離(或對稱差集的個數)差距大約在 1.15~1.19 倍之間。雖然我們未能以數學證明來保證我們的解距離最佳解真正的倍數，然而實驗數據顯示我們的解仍不失為一個好的結果。根據此數據，我們推斷當元素個數越多時，我們的解將會越逼近最佳解。所

以在實際的資料上（資料量很大時），我們的演算法會有不錯的執行效率且結果也會更接近實際的最佳解。

ρ	ratio(our heuristic solution/optimal solution)
10	1.1902
20	1.1786
30	1.1702
40	1.1611
50	1.1539

表 12. 2-PD 問題的模擬結果(找出最佳解與我們的解的差異)，針對 D(ρ,10)資料集。ratio 此為我們的解除以最佳解。第一個分割為 D(ρ,10)，其餘的分割為亂數產生，每個數據的產生是執行 100 組資料後的平均值，標準差為 0.0006。

4.3 網頁版本

最後一節，我們將實作我們的啟發式演算法的程式，並放到一個網站上。網址為: http://210.240.178.54 /kpc.html。這個網站是透過 PHP 程式碼完成的，裡面實作了我們的演算法，分別針對 k-PC 問題與 k-PD 問題，也包含了 Gusfield [8]的演算法（匈牙利演算法），來解決 2-PD 問題。我們可以很容易的知道 2-PD 問題的解乘 2 即為 2-PC 問題的解，所以這個網站就不提供 2-PC 問題的解（因為乘 2 既可得知）。 此網站我們提供了五種輸入資料的模式：（1）使用者模式(user mode)。按此模式，資料（元素）可由使用者自行輸入。對於每個分割 P_i($1 \le i \le k$)，我們依序的輸入元素(1 到 n)在 P_i 中所屬的群集。(2)亂數模式(random mode)。此模式中所有的分割由亂數產生（亂數是透過 PHP 的亂數產生器）。（3）D(ρ,1)模式。此

一模式即為第一個分割是 D(ρ,1)(ρ個群集，每個群集有一個元素)，其餘的分割為亂數產生(4) D(ρ,10)模式。此一模式即為第一個分割是 D(ρ,10)（(ρ個群集，每個群集有十個元素)），其餘的分割為亂數產生。（5）檔案模式(File mode)。此模式類似使用者模式，差別在於輸入元素的群集是透過檔案，即對於每個分割 P_i（$1 \le i \le k$），我們依序的輸入元素(1 到 n)在 P_i 中所屬的群集在一個文字檔，上傳後即可。我提供了三種演算法 (1)針對 k-PD 問題的啟發式演算法(2)針對 k-PC 問題的啟發式演算法（3）Gusfield 的演算法 [8]（會產生 2-PD 問題的最佳解）。這個網站一開始會要求輸入分割的個數(k);這 k 個分割中，群集個數最大的那個值（ρ）; 所有的群集中最多能有多少個元素個數（N_{max}），總元素的個數（n，n 必須小於等於ρ＊N_{max}）。然後選擇一種資料輸入的模式。之後按 submit，會跳到第二個網頁，此網頁要求輸入演算法的種類。任選其中一種，按下 Run，即可看到程式執行的結果，我們的網頁有做了相對的容錯輸入，如果輸入的資料有誤，會出現”You don't type data or input error”，要求重新輸入資料。此網站可供生物學家需要計算計算分割距離或找尋一致性分割時使用。

第五章 結論與未來研究方向

本研究探討兩個分割距離問題，並對此兩問題的提出啟發式演算法來加快其執行速度。我們提出以群集的特徵值進行排序，快速將群集對齊的演算法。以下是我們的結論：

首先，關於 k-PD 與 k-PC 問題的探討，從 2-PD 等價 2-PC 問題我們發現只要先對 k-PC 問題求解再將所得到的對稱差集移除，就可以得到 k-PD 問題的解。然後，我們提出了一個新的啟發式演算法，其執行的時間複雜度（即為 $O(k^2n+k\rho \log \rho)$）遠比 Berman 等人提出的 k-PC 的 2 倍近似演算法好（即為 $O(k^2(n+\rho^3))$），也比 k=2 時的最佳解好（$O(n+\rho^3)$）。我們的啟發式演算法是設計一特徵值代表每個分割中的每個群，兩個群的特徵值越接近則這兩群的對稱差集越小，透過特徵值的求取以及排序，可以快速對齊所有的 k 個分割。最後，透過實驗也證實了我們的想法。實驗結果顯示我們的演算法之執行效率上確實比最佳解的演算法快的多，我們的實驗也顯示出我們的解雖然不能達到最佳解，但差距並不算太多。

本研究提出的演算法在速度上確實符合預期。雖然本演算法對 k-PC 及 k-PD 問題能夠在合理時間內求出最佳解，然而仍有許多不同的分割問題，我們不能確定在其他的問題上也能有一樣的速度。此外，因為本研究主要聚焦在以一種新的方法使分割對齊，所以特徵值 s 的設計上並沒有太多的研究，而特徵值設計的好壞與演算法求解的精確性有密不可分的關係。總結以上，在未來的研究方向上我們建議可以針對如何應用本研究之演算法於不同的分割問題上，以及特徵值重新設計與改良，例如加入更多不同的因素或者使用現有的三種因素(群集中的元素個數、元素編號總和，元素編號間距)以不同的運算方式結合，以求提升演算法求解的精確度進而找到最佳解在 $k=2$ 時，或是提出一個證明的方式對我們演算法找出的解距離最佳解間有若干倍數的保證，當 $k>2$ 時。此外，本論以ρ=10，20，30，40，50 來進行實驗，在不同的ρ值下，實驗得到的解與最佳解的比(ratio)與ρ值之間看不出規則，例如 D(ρ,1)中在ρ=30 的情況下 ratio 為所有ρ中最低的，在ρ=40 時則是最高的，為何造成這種結果仍有待探討。

參考文獻

[1] A. Almudevar and C. Field, Estimation of single generation sibling relationships based on DNA markers, Journal of Agricultural, Biological, and Environmental Statistics **4** (1999), 136–165.

[2] A.M. Bagirov and K. Mardaneh, Modified global k-means algorithm for clustering in gene expression data sets, in Proceedings of the ACM 2006 workshop on Intelligent systems for bioinformatics **73** (2006), 23–28.

[3] A. Ben-Dor and Z. Yakhin, Clustering gene expression patterns, Journal of Computational Biology **6** (1999), 281–297.

[4] K. Butler, C. Field, C.M. Herbinger and B.R. Smith, Accuracy, efficiency and robustness of four algorithms allowing full sibship reconstruction from DNA marker data, Molecular Ecology **13** (2004), 1589–1600.

[5] P. Berman, B. DasGupta, M.-Y. Kao and J. Wang, On constructing an optimal consensus clustering from multiple clusterings, Information Processing Letters **104** (2007), 137–145.

[6] J. Beyer and B. May, A graph-theoretic approach to the partition of individuals into full-sib families, Molecular Ecology **12** (2003), 2243–2250.

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithm, 2nd edition, MIT Press, Cambridge, 2001.

[8] D. Gusfield, Partition-distance: A problem and class of perfect graphs arising in clustering, Information Processing Letters **82** (2002), 159–164.

[9] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Academic Press 2nd edition, San Francisco, 2006.

[10] D.A Konovalov, C. Manning and M.T. Henshaw, KinGroup: a program for pedigree relationship reconstruction and kin group assignments using genetic markers, Molecular Ecology Notes **4** (2004), 779–782.

[11] D.A Konovalov, B. Litow and N. Bajema, partition-distance via the assignment problem, Bioinformatics **21** (2005), 2463-2468

[12] S. Olafsson, X. Li and S. Wu, Operations research and data mining, European Journal of Operational Research **187** (2008), 1429–1448.

[13] C.H. Papadimitriou, Computational Complexity, Addison Wesley 1994.

[14] B. Saglam, F.S. Salman, S. Sayin and M. Turkay, A mixed-integer programming approach to the clustering problem with an application in customer segmentation, European Journal of Operational Research **173** (2006), 866–879.

[15] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

[16] A.J. Vakharia and J. Mahajan, The Clustering of objects and attributes for manufacturing and marketing applications, European Journal of Operational Research **123** (2000), 640–651.