2020 年臺灣國際科學展覽會
優勝作品專輯

作品編號 010034

參展科別 數學

作品名稱 改良式廣度優先網路爬蟲演算法之組合分析

得獎獎項 大會獎：一等獎

青少年科學獎

出國正選代表

就讀學校 臺北市立大直高級中學

指導教師 劉繕榜、胡裕仁

作者姓名 陳品劭

關鍵詞 *k*-錯排、遍歷、網路爬蟲

作者簡介

我是陳品劭，居於基隆，就讀臺北市立大直高級中學。我熱愛數學與資訊相關領域。

我對於資訊科技、資料科學等領域感興趣，而其背後的基礎即為數學。我於高中階段是針對離散數學中的廣義錯排與分散式網路爬蟲的關係進行探討。我熱愛進行數學研究並嘗試將其推廣應用至資訊領域中，即為藉由資訊輔助來研究數學如何應用至資訊。

摘要

本研究旨在探討分散式網路爬蟲瀏覽時間及覆蓋率之最佳化問題原理。藉由相異物排列所形成的循環組關係式進行一系列的探討。在 n 個相異元素的簡單排列中，不存在任意元素個數為 k ($k \leq n$)的子集對應到自己本身所成集合，我們稱此型態的排列方式為 ***k*-錯排**。換言之，假如 n 個相異元素進行簡單排列，排列後每個元素都不在原來的位置上，此時這樣的排列稱為一般的**錯排列**，也就是 1-錯排。本研究從分散網路爬蟲搜尋網址中進行相關發想，發現它的本質是遍歷完所有的頂點且沒有重複經過，即所謂「哈密頓路徑(Hamiltonian path)問題」中一筆畫的 NP-hard 問題，即圖遍歷問題的一種。因此本研究由 k-錯排遞迴之性質來探討分散式網路爬蟲最佳化問題。最後透過電腦模擬及組合數學分析推導，本研究將提出改善以 k-錯排應用至分散網路爬蟲的最佳化方式。

Abstract

This purpose of this study is to explore the principles with optimization of the browsing time and the most coverage of the web crawlers on the Webs. By the cyclic group relationship formed by the permutation of different elements do a series of discussions. In a simple permutation of n distinct elements, a permutation σ is k-derangement if for any subset X with cardinality k ($k \leq n$) such that $\sigma(X) \neq X$. In other words, if a permutation which leave no element fixed, known as derangements, then we say it is 1-derangement. We encode n distinct elements to be consecutive integers from 1 to n. After a simple permutation, we use algebraic group theory to express the correspondence, called cycle group. The number of cycle groups formed by k-derangement is called the number of web crawlers since we found that the cyclic group structure formed by k-derangement is related to integer partition.

In our study, from the scattered web crawler Search Web site, we discover its essence is to go through each web site exactly once, that is, the so-known "Hamiltonian path problem" an NP-hard problem, a special case of the travelling salesman problem. That is one of the graph traversal problems. We use the Breadth-First-Search algorithm to traverse, searching from a point in the graph, then searching for all adjacent and unvisited points at this point, and the searched nodes continue to search first and then deep. The nature of search is recursion. Therefore, our study explores the optimization of distributed web crawlers by the recursion nature of k-derangements. Finally, through computer simulation and combined mathematical analysis, this study successfully proposes to improve the optimal method of k-derangement application to distributed web crawlers.

壹、 前言

一、 研究動機

我們在高一數學課程上到排列組合單元[4]，老師特別推薦我們閱讀一本有關數學的書籍，其中有一篇的標題是『賽馬場的一天」[5]。故事情境是有十位學生玩一個叫－馬兒排排站的遊戲。遊戲是總共有四匹馬兒進行比賽，而十位學生要猜四匹馬兒的名次，而只要猜中一匹馬以上的名次就能得錢。在書中說明四匹馬兒的編號及它們的名次之間的對應關係就是一種排列。我們進一步延伸，考慮一個有n個元素的排列，若一個排列中所有的元素都不在自己原來的位置上，那麼這樣的排列就稱為原排列的一個*錯排*。研究一個排列錯排個數的問題，叫做錯排問題。這跟著名的帽子問題或拿名片的問題一樣[6]。

後來，我們在資訊課程學到有關網路爬蟲的概念。*網路爬蟲*是一種自動流覽全球資訊網的網路機器人，為搜尋引擎的重要組成部分，其目的為搜索網站，盡量不要搜索重覆的網站。這樣情形如同一個給定的圖，假設圖中兩點u, v間有一個點串列(a list of vertices)，在點串列中任意相鄰之點都有邊相連，即表示兩點間存在一條路徑[3,8]。在一個圖中，判斷是否存在著一條恰好通過所有的點，並且沒有重複經過的路徑。從這個觀點來看是一筆畫問題，也就是*哈密頓路徑*(Hamiltonian path)[3,8]─在圖中存在一條路徑通過且僅通過每一個頂點一次。而閉合的哈密頓路徑稱作*哈密頓迴路*(Hamiltonian cycle)[3,8]。

遍歷(Traversal)指的是在圖從某個點v出發開始，依照某種順序經由邊拜訪與已拜訪的點相鄰的點，最終拜訪過圖內所有點[3,8]。從網路爬蟲的觀點來看，搜索完特定區域的所有網站即表示遍歷完此區域。分散網路爬蟲主要是透過多個網路爬蟲遍歷全球資訊網。它主要是透過廣度優先進行搜索[11]，本質是一種圖形(graph)搜索演算法。從圖的某一點開始搜索，接著搜索此一點所有相鄰且未拜訪過的點，由搜索過的節點繼續進行先廣後深的搜尋。

我們注意到分散網路爬蟲的運作方式與錯排問題有些關連[12]。當我們利用電腦幫助我們進行錯排模擬時，我們發現排列後有一些循環組結構。例如：數字1, 2, 3, 4, 5進行排列時，1對應到2，2對應到3，3對應到1，4對應到5，5對應到4，這樣就形成兩個循環組，我們以(1 2 3)(4 5)表示。當我們尋找有關錯排列的相關文獻，發現已經有人針對一般錯排列的個數及循環組數進行研究[1,2]。後來我們找到文獻[13]提及2-錯排的遞迴關係式，引起我的好奇心。

但在文獻中僅有列出遞迴關係式的結果，而無提到證明過程。為想了解此問題，所以寄了一些email請教相關文獻作者，終於等到美國密蘇里州立大學Reid L.教授的回覆(personal communication, June 03, 2019)，得知其推導過程。我因而產生興趣嘗試推導k-錯排的遞迴關係式($k\geq 3$)，之後對於k-錯排問題多次向Reid L.教授請教，也對於文獻[13]中提出的開放問題進行研究。

回應到網路爬蟲的概念，我們希望網路搜尋能有效率地進行，就是探討有關網路爬蟲的個數及瀏覽時間的問題。我們將錯排應用至分散網路爬蟲上，也就是判斷它能否在最短時間遍歷完所有的網站而沒有重複經過。因此在研究中，我們將循環組數對應至網路爬蟲個數，並探討分散網路爬蟲覆蓋率及瀏覽時間之最佳化問題。也就是找出此時遍歷的哈密頓路徑路徑（一個環或者一條鏈）。如果在內部網路，由於路徑閉合（一個圈），則稱為哈密頓迴路。

二、 錯排相關名詞解釋及定義

(一) $m-cycle$(循環組)[2]

如圖1，上列代表著一數列，而下列表該數列每項所對應的數，用代數中群論的方法可表示為(135)(24)。其中(135)(1→3→5→1)為一個$3-cycle$，(24)(2→4→2)為一個$2-cycle$(如圖2)。

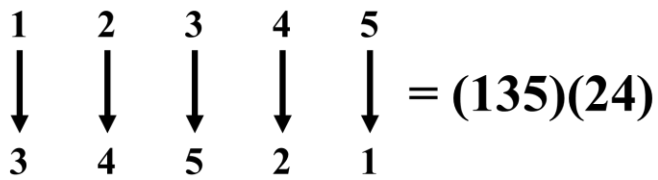

圖 1 $m-cycle$矩陣表示法

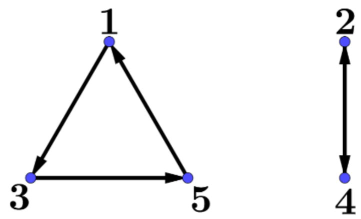

圖 2 $m-cycle$圖形表示法

(二) k-錯排[7]

設S_n表n個相異元素(假設為1, 2, …, n連續整數)所有排列所成集合。若$\sigma\in S_n$且對所有的$X\subseteq\{1,2,\cdots,n\}$且$|X|=k$，滿足$\sigma(X)\neq X$，我們稱$\sigma$為*k-錯排*。所有$k$-錯排所形成的集合以$D_{k,n}$表示[7]。假如$|X|=1$，也就是$n$個相異元素進行簡單排列，排列後每個元素都不在原來的位置上，此時這樣的排列稱為一般的錯排。錯排又稱不定點排列，即不產生$1-cycle$，也就是1-錯排。例如：$D_{1,4}$為4個相異元素所有排列中，1-錯排所成集合，即

$D_{1,4}=\{(1234),(1243),(1324),(1342),(1432),(1423),(12)(34),(13)(24),(14)(23)\}$ 。

三、 網路爬蟲相關名詞解釋

(一) 分散網路爬蟲[9]

一種用來自動瀏覽全球資訊網的網路機器人，其目的一般為編纂網路索引及網路搜尋引擎等。透過爬蟲更新自身的網站內容或其對其他網站的索引，它可以將自己所存取的頁面儲存下來，以便搜尋引擎事後生成索引供用戶搜尋。分散網路爬蟲主要為透過多個網路爬蟲遍歷全球資訊，如圖3所示。

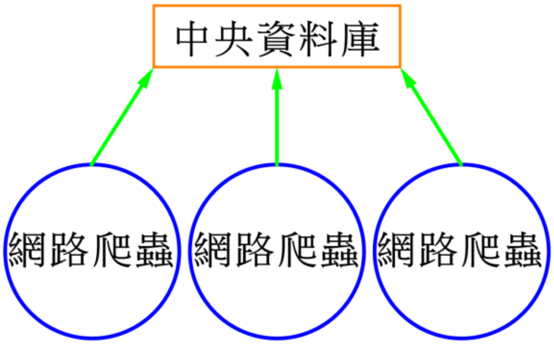

圖 3 分散式網路爬蟲

(二) 爬行疆域與圖的對應

網路爬蟲進行搜索時，會先編纂出頁面上所有的超連結，並將它們寫入一張「待訪列表」，即所謂*爬行疆域*，並依照此表得順序進行搜索[9]。在研究中我們透過圖論方法來探討。圖是由一些頂點和邊所組成的結構。設圖G所有頂點所成的集合為$V(G)$，而圖G中所有邊所成集合為$E(G)$。若圖G中有n個頂點且任兩點都有邊相連，我們就稱圖G為n-完全圖，以K_n表示。若圖H及圖G滿足如果$V(H)\subseteq V(G)$且$E(H)\subseteq E(G)$，我們稱H為G的子圖[3]，如圖5就是圖4的一個子圖。

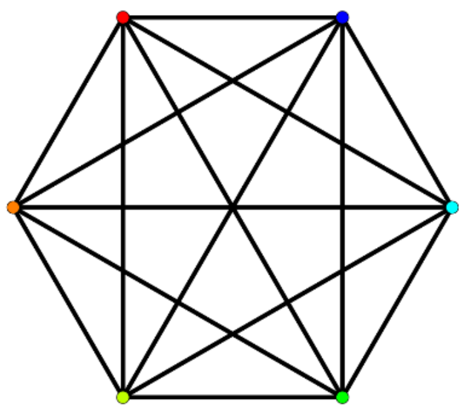

圖4 完全圖 K_6 (網站間連結情形)

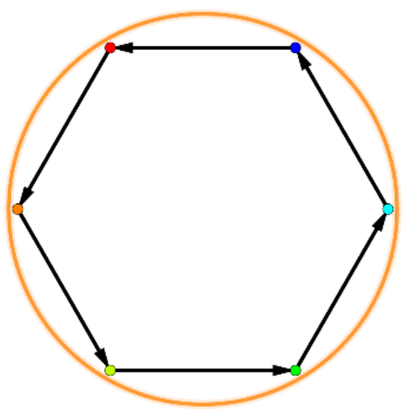

圖5 K_6的子圖(其中一種爬行疆域)

我們設一個網站為一個點，而一個URL為連結兩個網站(點)的邊(我們以完全圖 K_n 為例，如圖4即為 K_6)。圖形中的一個子圖，即為一種爬行疆域。圖中所產生的每個*cycle*皆各別由一隻網路爬蟲負責，因此循環組數可以對應成網路爬蟲個數(如圖5)。

四、 研究目的

我們期望以組合的觀點進行分析。並應用至網路爬蟲的最佳化上。基於此，本研究的目的如下。

(一) 探討k-錯排相關性質。

1. 推導$\left|D_{k,n}\right|$的遞迴關係式。
2. 推導於 k-錯排下循環組數的遞迴關係式。
3. 提出 k-錯排演算法。
4. k-錯排數值分析。
5. 文獻[13]開放問題：$\left|D_{k,n}\right| \equiv 0 \pmod k$。

(二) 分析網路爬蟲最佳化問題。

1. 分析網路爬蟲的遍歷行為。
2. 提出網路爬蟲最佳爬行疆域。
3. 如何改良分散式網路爬蟲？

五、 研究設備及器材

硬體部分：紙、筆、電腦。

軟體部分：C語言、幾何軟體Geogebra、文書處理軟體Word、方程式編輯器Mathtype。

貳、 研究方法或過程

一、 研究架構圖

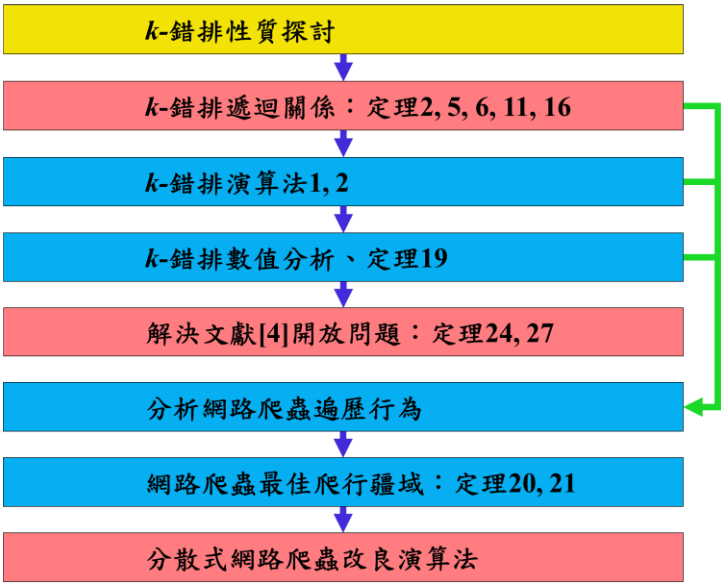

圖 6 研究架構圖

二、 *k*-錯排數的遞迴關係式

(一) $\left|D_{2,n}\right|$ 的遞迴關係式

文獻[2,5]指出一般的錯排數 $\left|D_{1,n}\right|$ 的遞迴關係式，也就是

$$\left|D_{1,n}\right|=(n-1)\left(\left|D_{1,(n-1)}\right|+\left|D_{1,(n-2)}\right|\right)\quad(n\geq 3)$$

其中 $\left|D_{1,1}\right|=0$，$\left|D_{1,2}\right|=1$。

且文獻[13]中指出2-錯排數 $\left|D_{2,n}\right|$ 的遞迴關係式，也就是

$$\begin{aligned}\left|D_{2,n}\right|=&\left|D_{2,(n-1)}\right|+(n-1)(n-3)\left|D_{2,(n-2)}\right|+(n-1)(n-2)\left|D_{2,(n-3)}\right|\\&+(n-1)(n-2)(n-3)\left|D_{2,(n-4)}\right|\quad(n\geq 4)\end{aligned}$$

其中 $\left|D_{2,0}\right|=1$，$\left|D_{2,1}\right|=1$，$\left|D_{2,2}\right|=0$，$\left|D_{2,3}\right|=2$。

因此我們猜測是否有 $\left|D_{3,n}\right|$ 的遞迴關係式甚至是 $\left|D_{k,n}\right|$ 的遞迴關係式。

定義1：S_n 表 n 個相異元素(假設為 $1,2,\ldots,n$ 連續整數)所有排列所成集合，則所有同時符合一般的錯排和 k-錯排所形成的集合以 $d_{k,n}$ 表示。其中 $d_{1,n}=D_{1,n}$。

本研究中，將透過引理1求得與文獻[13]不同形式的$\left|D_{2,n}\right|$遞迴關係式，即定理2。

由於本研究與研究[2]中一般錯排的遞迴關係式證明所使用的手法類似，所以特別提出並和本研究中2-錯排數$\left|D_{2,n}\right|$(定理2)做個比較。

引理 1[2]：$\left|d_{2,n}\right|=(n-1)\left(\left|d_{2,n-1}\right|+(n-2)\left|d_{2,n-3}\right|\right)\quad(n\geq 4)$，其中$\left|d_{2,1}\right|=0$，$\left|d_{2,2}\right|=0$，$\left|d_{2,3}\right|=2$。

證明：

根據文獻[2]，n個相異物進行排列，在符合「一般錯排」情形下，不產生$2-cycle$的排列數，即為同時不產生$1-cycle$、$2-cycle$，也就表示同時符合 1-錯排、2-錯排，也就等於$\left|d_{2,n}\right|$。當n個相異物進行排列，1 排在 2，則剩餘的即為$\left|d_{2,n-1}\right|$，而因為不能產生 2-錯排，所以 2 不能排 1，所以排在 3，而 3 排回 1，剩餘即為$\left|d_{2,n-3}\right|$，而 2 有$n-2$排法，故再乘以$n-2$，而 1 有$n-1$種排法，故再乘以$(n-1)$，可得$d_{2,n}$具有以下遞迴關係式：

$$\left|d_{2,n}\right|=(n-1)\left(\left|d_{2,n-1}\right|+(n-2)\left|d_{2,n-3}\right|\right)\quad(n\geq 4)$$

其中$\left|d_{2,1}\right|=0$，$\left|d_{2,2}\right|=0$，$\left|d_{2,3}\right|=2$。 □

定理 2：$\left|D_{2,n}\right|=\left|d_{2,n}\right|+n\times\left|d_{2,n-1}\right|\quad(n\geq 2)$，其中$\left|D_{2,1}\right|=1$。

證明：

$\left|D_{2,n}\right|$表不產生$2-cycle$的排列數。當n個相異物進行排列，不形成$2-cycle$或兩個$1-cycle$的排列可分為兩種情形。一是沒有形成$1-cycle$的排列，另一是只有一個$1-cycle$的排列。而同時不產生$1-cycle$、$2-cycle$的排列數即為$\left|d_{2,n}\right|$，而形成$1-cycle$的排列但不產生$2-cycle$的排列，我們將 1 排在 1，形成一個$1-cycle$，而剩下$n-1$個元素不能產生$1-cycle$，也不能產生$1-cycle$，即為$\left|d_{2,n-1}\right|$，而形成$1-cycle$的元素有n個可能，因此再乘以n，即可得$\left|D_{2,n}\right|$的遞迴關係式$\left|D_{2,n}\right|=\left|d_{2,n}\right|+n\times\left|d_{2,n-1}\right|\quad(n\geq 2)$，其中$\left|D_{2,1}\right|=1$。 □

(二) $\left|D_{3,n}\right|$的遞迴關係式

我們利用$\left|D_{2,n}\right|$的推導方式來去推導。$\left|D_{3,n}\right|$即為符合3-錯排的排列數，而根據文獻

[2]，n人在「錯排」下亦不能產生$3-cycle$的方法數，存在遞迴關係式。而此情況也剛好可以等同於同時符合1-錯排、3-錯排的方法數，因為不產生$1-cycle$、$3-cycle$，即不會產生不符合3-錯排的排列情況：$(3),(2)(1),(1)(1)(1)$。

引理 3[2]：$|d_{3,n}|=(n-1)\left(|d_{3,n-2}|+|d_{3,n-1}|-(n-2)|d_{3,n-3}|+(n-2)(n-3)|d_{3,n-4}|\right)\quad(n\geq5)$

其中$|d_{3,1}|=0$，$|d_{3,2}|=1$，$|d_{3,3}|=0$，$|d_{3,4}|=9$。

證明：略 □

但$|d_{3,n}|$中可能包含$2-cycle$，仿照$|D_{2,n}|$的遞迴關係式先排一個$1-cycle$的話，會計算到不符合 3-錯排的排列，對此我們又於文獻[2]中找到解決的方法。

定義 2：n 個相異元素於一般錯排下排列，產生 j 組循環組數，且每組循環組至少 i 人的方法數，記作$F(n,i,j)$。

引理 4[2]：$F(n,i,j)=(n-1)F(n-1,i,j)+\left(\prod_{k=1}^{i-1}(n-k)\right)F(n-i,i,j-1)$，

其中$F(n,i,j)=0\quad(n<i\times j)$，$F(n,i,1)=(n-1)!\quad(n\geq i)$。

證明：略 □

由$F(n,i,j)$即可求得同時不產生$1-cycle$、$2-cycle$、$3-cycle$的排列數，即為$\sum_{j=1}^{n}F(n,4,j)$，據此可推論出$|D_{3,n}|$的遞迴關係式。

定理 5：$|D_{3,n}|=|d_{3,n}|+C_1^n\times\sum_{j=1}^{n-1}F(n-1,4,j)+C_2^n\times\sum_{j=1}^{n-2}F(n-2,4,j)\quad(n\geq3)$，

其中$|D_{3,1}|=1$, $|D_{3,2}|=2$。

證明：

n 個相異物進行排列，不產生$3-cycle$或一個$2-cycle$和一個$1-cycle$或三個$1-cycle$的排列可分為，同時也不產生$1-cycle$、$2-cycle$、$3-cycle$的排列，或產生$1-cycle$但不產生兩個$1-cycle$、$2-cycle$、$3-cycle$的排列，或產生兩個$1-cycle$但不產生三個$1-cycle$、$2-cycle$、$3-cycle$的排列。而同時不產生$1-cycle$、$2-cycle$、$3-cycle$的排列數即為$|d_{3,n}|$。而產生$1-cycle$，

但不產生兩個$1-cycle$、$2-cycle$、$3-cycle$的排列，我們將 1 排在 1，形成一個$1-cycle$，而剩下$n-1$個元素不能產生$1-cycle$、$2-cycle$、$3-cycle$，即為$\sum_{j=1}^{n-1} F(n-1,4,j)$，而自己排自己產生一個$1-cycle$的可能有$C_1^n$個，因此再乘以$C_1^n$。而產生兩個$1-cycle$，但不產生三個$1-cycle$、$2-cycle$、$3-cycle$的排列，我們將 1 排在 1，2 排在 2，形成一個$1-cycle$，而剩下$n-2$個元素不能產生$1-cycle$、$2-cycle$、$3-cycle$，即為$\sum_{j=1}^{n-2} F(n-2,4,j)$，而自己排自己產生兩個$1-cycle$的可能有$C_2^n$個，因此再乘以$C_2^n$，即得到以下遞迴關係式

$$\left|D_{3,n}\right| = \left|d_{3,n}\right| + C_1^n \times \sum_{j=1}^{n-1} F(n-1,4,j) + C_2^n \times \sum_{j=1}^{n-2} F(n-2,4,j) \quad (n \ge 3)，$$

其中$\left|D_{3,1}\right| = 1, \left|D_{3,2}\right| = 2$。 □

(三) $\left|D_{4,n}\right|$的遞迴關係式

定理 6：

$$\left|D_{4,n}\right| = \left|d_{4,n}\right| + C_1^n \times \sum_{j=1}^{n-1} F(n-1,5,j) + C_2^n \times \sum_{j=1}^{n-2} F(n-2,5,j) + C_3^n \times \sum_{j=1}^{n-3} F(n-3,5,j)$$
$$+ n \times C_2^{n-1} \times \sum_{j=1}^{n-3} F(n-3,5,j) \quad (n \ge 4)$$

，其中$\left|D_{4,1}\right| = 1,\ \left|D_{4,2}\right| = 2,\ \left|D_{4,3}\right| = 6$。

證明：

$\left|D_{4,n}\right|$表示為 n 個相異物進行直線排列，其中符合 4-錯排的排列。而符合 4-錯排的情況可分為不產生$1-cycle$、產生 1~3 個$1-cycle$、產生 1 個$1-cycle$c 和一個$2-cycle$。不產生$1-cycle$並符合 4-錯排的情況，即為$\left|d_{4,n}\right|$。產生 1 個$1-cycle$的情況，有 n 個數可以選，故為C_1^n，而$n-1$個相異物進行排列，每組循環組至少包含 5 個相異物即為$\sum_{j=1}^{n-1} F(n-1,5,j)$。產生 2 個$1-cycle$的情況，有 n 個數可以選，故為C_2^n，而$n-2$個相異物進行排列，每組循環組至少包含 5 個相異物即為$\sum_{j=1}^{n-2} F(n-2,5,j)$。產生 3 個$1-cycle$的情況，有 n 個數可以選，故為C_3^n，而$n-3$

個相異物進行排列，每組循環組至少包含 5 個相異物即為$\sum_{j=1}^{n-3} F(n-3,5,j)$。產生 1 個$1-cycle$和一個$2-cycle$的情況，有 n 個數可以選，故為$C_1^n \times C_2^{n-1}$，而$n-3$個相異物進行排列，每組循環組至少包含 5 個相異物即為$\sum_{j=1}^{n-3} F(n-3,5,j)$。故$|D_{4,n}|$的遞迴關係式如下：

$$
\begin{aligned}
|D_{4,n}| &= |d_{4,n}| + C_1^n \times \sum_{j=1}^{n-1} F(n-1,5,j) + C_2^n \times \sum_{j=1}^{n-2} F(n-2,5,j) + C_3^n \times \sum_{j=1}^{n-3} F(n-3,5,j) \\
&\quad + n \times C_2^{n-1} \times \sum_{j=1}^{n-3} F(n-3,5,j) \quad (n \geq 4)
\end{aligned}
$$

，其中$|D_{4,1}|=1, |D_{4,2}|=2, |D_{4,3}|=6$。 □

三、 2-錯排循環組數

定義 3：n 個相異元素(假設為 $1,2,\ldots,n$ 連續整數)進行直線排列，而符合 k-錯排且包含 j 組循環組的方法數記作$F_k(n,j) \quad (n \in \mathbb{N})$。

顯然地，$F_k(n,1)=(n-1)!\ (n \in \mathbb{N}, n \neq k)$。

文獻[2]中，有對於**一般錯排**下，也就是1-錯排下，產生j組循環組的方法數，進行探討，如下引理7所述。

引理 7[2]：$F_1(n,j)=(n-1)F_1(n-2,j-1)+(n-1)F_1(n-1,j)$。

證明：略 □

因此我們好奇於 2-錯排下，產生 j 組循環組的方法數，甚至是於 k-錯排下，產生產生 j 組循環組的方法數。

(一) $F_2(n,2)$的遞迴關係式

定理 8：$F_2(n,2)=C_1^n \times F_2(n-1,1)+F(n,3,2) \quad (n \geq 3)$，其中$F_2(1,2)=0$，$F_2(2,2)=0$。

證明：

$F_2(n,2)$為 n 個相異元素排列下，符合 2-錯排，且只產生 2 組循環組的方法數。符合 2-錯排的排列，由先前$|D_{2,n}|$的遞迴關係式推導時可得知，可分為產生 1 個$1-cycle$和不產生任何$1-cycle$。$F_2(n-1,1)$表示為$n-1$個相異物於符合 2-錯排下排列，只產生 1 組循環組的方法數，

而現在要多放 1 個元素進去，要產生 2 個循環組，故此元素要自行形成$1-cycle$，有 n 個元素可以選擇形成$1-cycle$，故再乘C_1^n。剩下的是不產生任何$1-cycle$、$2-cycle$的方法數，且產生 2 個循環組的方法數，即為$F(n,3,2)$。故$F_2(n,2)=C_1^n\times F_2(n-1,1)+F(n,3,2)\quad(n\ge3)$，其中$F_2(1,2)=0$, $F_2(2,2)=0$。 □

(二) $F_2(n,3)$的遞迴關係式

定理 9：$F_2(n,3)=C_1^n\times F(n-1,3,2)+F(n,3,3)\quad(n\ge2)$，其中$F_2(1,3)=0$。

證明：

$F_2(n,3)$為 n 個相異元素排列下，符合 2-錯排，且只產生 3 組循環組的方法數。依照$F_2(n,2)$的推導方式來進行推導，一樣分成產生 1 個$1-cycle$和不產生任何$1-cycle$。但$F_2(n-1,2)$表示為$n-1$個相異物於符合 2-錯排下排列，只產生 2 組循環組的方法數，可能會產生$1-cycle$，使得其不符合 2-錯排，我們因而改用$F(n-1,3,2)$，表示為$n-1$個相異物進行排列，只產生 2 組循環組，且每組循環組至少 3 個相異物的方法數，而現在要多放 1 個元素進去，要產生 3 個循環組，故此元素要自行形成$1-cycle$，有 n 個元素可以選擇形成$1-cycle$，故再乘C_1^n。剩下的是不產生任何$1-cycle$、$2-cycle$，且產生 3 個循環組的方法數，即為$F(n,3,3)$。故$F_2(n,3)=C_1^n\times F(n-1,3,2)+F(n,3,3)\quad(n\ge2)$，其中$F_2(1,3)=0$。 □

(三) $F_2(n,4)$的遞迴關係式

定理 10：$F_2(n,4)=C_1^n\times F(n-1,3,3)+F(n,3,4)\quad(n\ge2)$，其中$F_2(1,4)=0$。

證明：

$F_2(n,4)$為 n 個相異元素排列下，符合 2-錯排，且只產生 4 組循環組的方法數。依照$F_2(n,3)$的推導方式來進行推導，一樣分成產生 1 個$1-cycle$和不產生任何$1-cycle$。$F(n-1,3,3)$表示為$n-1$個相異物進行排列，只產生 3 組循環組，且每組循環組至少 3 個相異物的方法數，而現在要多放 1 個元素進去，要產生 4 個循環組，故此元素要自行形成$1-cycle$，有 n 個元素可以選擇形成$1-cycle$，故再乘C_1^n。剩下的是不產生任何$1-cycle$、$2-cycle$，且產生 4 個循環組

的方法數，即為 $F(n,3,4)$。故 $F_2(n,4)$ 的遞迴關係式如下：

$$F_2(n,4)=C_1^n\times F(n-1,3,3)+F(n,3,4)\quad(n\ge 2)，$$

其中 $F_2(1,4)=0$。 □

(四) $F_2(n,j)$ 的遞迴關係式

定理 11：$F_2(n,j)=C_1^n\times F(n-1,3,j-1)+F(n,3,j)\quad(n\ge 2, j\ge 3)$

證明：

$F_2(n,j)$ 為 n 個相異元素排列下，符合 2-錯排，且只產生 j 組循環組的方法數。由於 $F_2(n,4)$ 可依照 $F_2(n,3)$ 的推導方式來進行推導，因此我們猜測 $F_2(n,j)$ 一樣可以。同樣分成產生 1 個 $1-cycle$ 和不產生任何 $1-cycle$。$F(n-1,3,j-1)$ 表示為 $n-1$ 個相異物進行排列，只產生 $j-1$ 組循環組，且每組循環組至少 3 個相異物的方法數，而現在要多放 1 個元素進去，要產生 4 個循環組，故此元素要自行形成 $1-cycle$，有 n 個元素可以選擇形成 $1-cycle$，故再乘 C_1^n。剩下的是不產生任何 $1-cycle$、$2-cycle$，且產生 j 個循環組的方法數，即為 $F(n,3,j)$。故 $F_2(n,j)$ 的遞迴關係式為 $F_2(n,j)=C_1^n\times F(n-1,3,j-1)+F(n,3,j)\quad(n\ge 2, j\ge 3)$。 □

四、 3-錯排循環組數

(一) $F_3(n,2)$ 的遞迴關係式

定理 12：$F_3(n,2)=C_1^n\times F_3(n-1,1)+C_2^n\times F_3(n-2,1)+F(n,4,2)\quad(n\ge 5)$，其中 $F_3(1,2)=0$，

$F_3(2,2)=1$，$F_3(3,2)=0$，$F_3(4,2)=3$。

證明：

$F_3(n,2)$ 為 n 個相異元素排列下，符合 3-錯排，且只產生 2 組循環組的方法數。符合 3-錯排的排列，即為不能產生任意相加為 3 的循環組結構，分別有以下幾種情況，只產生 ≥ 4 的循環組、產生 1~2 個 $1-cycle$、產生 1 個 $2-cycle$。只產生 ≥ 4 的循環組，且只產生 2 組循環組即為 $F(n,4,2)$。產生 1 個 $1-cycle$ 的情況，有 n 個數可以做為那一個 $1-cycle$，故為 C_1^n，而

現在要產生 2 個循環組，$F_3(n-1,1)$表示為$n-1$個相異物於符合 3-錯排下排列，只產生 1 個循環組的方法數。而要產生 2 個$1-cycle$的情況，又只能產生 2 組循環組的情況只有$(1)(1)$，故於此無須考慮此情況。剩下要產生 1 個$2-cycle$的情況，產生$2-cycle$的方法數為C_2^n，其中$n-2$個相異物於符合 3-錯排下排列，只產生 1 組循環組的方法數為$F_3(n-2,1)$，故$F_3(n,2)$的遞迴關係式如下：

$$F_3(n,2)=C_1^n\times F_3(n-1,1)+C_2^n\times F_3(n-2,1)+F(n,4,2)\quad (n\geq 5)$$

其中$F_3(1,2)=0$，$F_3(2,2)=1$，$F_3(3,2)=0$，$F_3(4,2)=3$。 □

(二) $F_3(n,3)$的遞迴關係式

定理 13：

$$F_3(n,3)=C_1^n\times F(n-1,4,2)+C_2^n\times F(n-2,4,1)+C_2^n\times F(n-2,4,2)+C_2^n\times C_2^{n-2}\times\frac{1}{2!}\times F(n-4,4,1)$$
$$+F(n,4,3)\quad (n\geq 7)$$

，其中$F_3(n,3)=0\quad (1\leq n\leq 5)$，$F_3(6,3)=105$。

證明：

$F_3(n,3)$為 n 個相異元素排列下，符合 3-錯排，且只產生 3 組循環組的方法數。$F_3(n,3)$可以分成只產生≥ 4的循環組、產生 1~2 個$1-cycle$、產生 1~2 個$2-cycle$來進行討論。只產生≥ 4的循環組，且只產生 3 組循環組即為$F(n,4,3)$。產生 1 個$1-cycle$的情況，有 n 個數可以做為那一個$1-cycle$，故為C_1^n，而現在要產生 3 個循環組，$F(n-1,4,2)$表示為$n-1$個相異物進行排列，只產生 2 個循環組，且每組循環組至少 4 個相異物的方法數。產生 2 個$1-cycle$的情況，有 n 個數可以做為那 2 個$1-cycle$，故為C_2^n，而現在要產生 3 個循環組，$F(n-2,4,1)$表示為$n-2$個相異物進行排列，只產生 1 個循環組，且至少 4 個相異物的方法數。產生 1 個$2-cycle$的情況，產生$2-cycle$的方法數為C_2^n，其中$n-2$個相異物進行排列，只產生 2 個循環組，且每組循環組至少 4 個相異物的方法數為$F(n-2,4,2)$。產生 2 個$2-cycle$的情況，產生$2-cycle$的方法數為$C_2^n\times C_2^{n-2}\times\frac{1}{2!}$，其中$n-4$個相異物進行排列，只產生 1 個循環組，且至少

4 個相異物的方法數為 $F(n-4,4,1)$。故 $F_3(n,3)$ 的遞迴關係式如下：

$$F_3(n,3)=C_1^n\times F(n-1,4,2)+C_2^n\times F(n-2,4,1)+C_2^n\times F(n-2,4,2)+C_2^n\times C_2^{n-2}\times\frac{1}{2!}\times F(n-4,4,1)$$
$$+F(n,4,3)\quad(n\ge 7)$$

其中 $F_3(n,3)=0\quad(1\le n\le 5)$，$F_3(6,3)=105$。 □

(三) $F_3(n,4)$ 的遞迴關係式

定理 14：

$$F_3(n,4)=C_1^n\times F(n-1,4,3)+C_2^n\times F(n-2,4,2)+C_2^n\times F(n-2,4,3)+C_2^n\times C_2^{n-2}\times\frac{1}{2!}\times F(n-4,4,2)$$
$$+C_2^n\times C_2^{n-2}\times C_2^{n-4}\times\frac{1}{3!}\times F(n-4,4,1)+F(n,4,4)\quad(n\ge 9)$$

其中 $F_3(n,3)=0\quad(1\le n\le 7)$，$F_3(8,2)=105$。

證明：

$F_3(n,4)$ 為 n 個相異元素排列下，符合 3-錯排，且只產生 4 組循環組的方法數。$F_3(n,4)$ 可以分成只產生≥4 的循環組、產生 1~2 個 $1-cycle$、產生 1~3 個 $2-cycle$ 來進行討論。只產生≥4 的循環組，且只產生 4 組循環組即為 $F(n,4,4)$。產生 1 個 $1-cycle$ 的情況，有 n 個數可以做為那一個 $1-cycle$，故為 C_1^n，而現在要產生 4 個循環組，$F(n-1,4,3)$ 表示為 $n-1$ 個相異物進行排列，只產生 3 個循環組，且每組循環組至少 4 個相異物的方法數。產生 2 個 $1-cycle$ 的情況，有 n 個數可以做為那 2 個 $1-cycle$，故為 C_2^n，而現在要產生 4 個循環組，$F_3(n-2,4,2)$ 表示為 $n-2$ 個相異物進行排列，只產生 2 個循環組，且至少 4 個相異物的方法數。產生 1 個 $2-cycle$ 的情況，產生 $2-cycle$ 的方法數為 C_2^n，其中 $n-2$ 個相異物進行排列，只產生 3 個循環組，且每組循環組至少 4 個相異物的方法數為 $F_3(n-2,4,3)$。產生 2 個 $2-cycle$ 的情況，產生 $2-cycle$ 的方法數為 $C_2^n\times C_2^{n-2}\times\frac{1}{2!}$，其中 $n-4$ 個相異物進行排列，只產生 2 個循環組，且至少 4 個相異物的方法數為 $F_3(n-4,4,2)$，產生 3 個 $2-cycle$ 的情況，產生 $2-cycle$ 的方法數為 $C_2^n\times C_2^{n-2}\times C_2^{n-4}\times\frac{1}{3!}$，其中 $n-4$ 個相異物進行排列，只產生 1 個循環組，且至少 4 個相異物的

方法數為 $F_3(n-4,4,1)$。故 $F_3(n,4)$ 的遞迴關係式如下：

$$F_3(n,4)=C_1^n\times F(n-1,4,3)+C_2^n\times F(n-2,4,2)+C_2^n\times F(n-2,4,3)+C_2^n\times C_2^{n-2}\times\frac{1}{2!}\times F(n-4,4,2)$$

$$+C_2^n\times C_2^{n-2}\times C_2^{n-4}\times\frac{1}{3!}\times F(n-4,4,1)+F(n,4,4)\quad(n\geq 9)$$

其中 $F_3(n,3)=0\quad(1\leq n\leq 7)$，$F_3(8,2)=105$。 □

(四) $F_3(n,j)$ 的遞迴關係式

$F_3(n,j)$ 為 n 個相異元素排列下，符合 3-錯排，且只產生 j 組循環組的方法數。由定理 13、14 可以得知，隨著 j 增加，$2-cycle$ 可討論的情況增加，準確來說是與 k 互質且小於 k 的 $m-cycle$ 情況與 j 呈現正相關。舉例來說，當 $n=10$ 且 $j=3$ 時，最多也就出現產生 1 或 2 組 $2-cycle$ 的兩種情況 $(2)(4)(4),(2)(2)(6)$。但當 $j=4$ 時，則需考慮產生 3 組 $2-cycle$ 的情況 $(2)(2)(2)(4)$。因為與 k 互質的 m 乘以任何數都不會等於 k，因此 $m-cycle$ 可以隨著 j 增加，增加討論的情況。為處理此問題，我們採用遞迴方式提出定理 15。

定義 4：n 個相異元素(假設為 1, 2, …, n 連續整數)進行直線排列，而符合一般錯排(1-錯排)且包含 j 組循環組而不產生 $m-cycle$ 的方法數記作 $G(n,m,j)$。

定理 15：

$$G(n,m,j)=(n-1)G(n-2,m,j-1)+(n-1)G(n-1,m,j)-\left(\prod_{k=1}^{m-1}(n-k)\right)G(n-m,m,j-1)$$

$$+\left(\prod_{k=1}^{m}(n-k)\right)G(n-m-1,m,j-1)$$

證明：

n 個相異物進行直線排列，當 1 排在 k 而 k 排在 1 時即形成 1 個 $2-cycle$，而 1 有 $n-1$ 種選擇，且剩下的即為 $G(n-2,m,j-1)$。而 k 不排在 1 的情況則為 $G(n-1,m,j)$，但其中會產生 $m-cycle$ 的情況數需扣除，先排出一個 $m-cycle$ 的情況為 $\prod_{k=1}^{m-1}(n-k)$，再處理剩下的即為 $G(n-m,m,j-1)$。而 $G(n-1,m,j)$ 的情形中產生 $m-cycle$ 的情況會在 $G(n,m,j)$ 中會變為 $(m+1)-cycle$，因此需加回來，先排出一個 $m-cycle$ 的情況為 $\prod_{k=1}^{m}(n-k)$，再處理剩下的即為

$G(n-m-1,m,j-1)$。故 $G(n,m,j)$ 的遞迴關係式如下：

$$G(n,m,j)=(n-1)G(n-2,m,j-1)+(n-1)G(n-1,m,j)-\left(\prod_{k=1}^{m-1}(n-k)\right)G(n-m,m,j-1)$$
$$+\left(\prod_{k=1}^{m}(n-k)\right)G(n-m-1,m,j-1)$$

□

我們透過定理 15 提出 $F_3(n,j)$ 的遞迴關係式(定理 16)。

定理 16：$F_3(n,j)=C_1^n\times F(n-1,4,j-1)+C_2^n\times F(n-2,4,j-2)+G(n,3,j) \quad (n,j\geq 3)$

證明：

$F_3(n,4)$ 為 n 個相異元素排列下，符合 3-錯排，且只產生 4 組循環組的方法數。$F_3(n,4)$ 可以分成產生 0~2 個 $1-cycle$。由於定理 15 我們可以不用再討論 $2-cycle$ 的情況，而沒有產生 $1-cycle$ 的情況即為 $G(n,3,j)$。產生 1 個 $1-cycle$ 的情況，有 n 個數可以做為那一個 $1-cycle$，故為 C_1^n，而現在要產生 j 個循環組，$F(n-1,4,j-1)$ 表示為 $n-1$ 個相異物進行排列，只產生 $j-1$ 個循環組，且每組循環組至少 4 個相異物的方法數。產生 2 個 $1-cycle$ 的情況，有 n 個數可以做為那 2 個 $1-cycle$，故為 C_2^n，而現在要產生 j 個循環組，$F_3(n-2,4,2)$ 表示為 $n-2$ 個相異物進行排列，只產生 j-2 個循環組，且每組循環組至少 4 個相異物的方法數。故

$$F_3(n,j)=C_1^n\times F(n-1,4,j-1)+C_2^n\times F(n-2,4,j-2)+G(n,3,j) \quad (n,j\geq 3)$$

□

五、 4-錯排循環組數

(一) $F_4(n,2)$ 的遞迴關係式

定理 17：

$F_4(n,2)=C_1^n\times F(n-1,5,1)+C_2^n\times F(n-2,5,1)+C_3^n\times F(n-3,5,1)+F(n,5,2) \quad (n\geq 8)$，其中 $F_4(1,2)=0$，$F_4(2,2)=1$，$F_4(3,2)=3$，$F_4(4,2)=0$，$F_4(5,2)=20$，$F_4(6,2)=184$，$F_4(7,2)=1344$。

證明：

$F_4(n,2)$為 n 個相異元素排列下，符合 4-錯排，且只產生 2 組循環組的方法數。可以分為以下幾種情況，只產生≥ 5的循環組、產生 1 個$1-cycle$、產生 1 個$2-cycle$、產生 1 個$3-cycle$。只產生≥ 5的循環組，且只產生 2 組循環組即為$F(n,5,2)$。產生 1 個$1-cycle$的情況，有 n 個數可以做為那一個$1-cycle$，故為C_1^n，而現在要產生 2 個循環組，$F(n-1,5,1)$表示為$n-1$個相異物進行排列，只產生 1 個循環組，且每組循環組至少 5 個相異物的方法數。產生 1 個$2-cycle$的情況，產生$2-cycle$的方法數為C_2^n，而現在要產生 2 個循環組，$F(n-2,5,1)$表示為$n-2$個相異物進行排列，只產生 1 個循環組，且每組循環組至少 5 個相異物的方法數。產生 1 個$3-cycle$的情況，產生$3-cycle$的方法數為C_3^n，而現在要產生 2 個循環組，$F(n-3,5,1)$表示為$n-3$個相異物進行排列，只產生 1 個循環組，且每組循環組至少 5 個相異物的方法數。故$F_4(n,2)$的遞迴關係式如下：

$$F_4(n,2)=C_1^n\times F(n-1,5,1)+C_2^n\times F(n-2,5,1)+C_3^n\times F(n-3,5,1)+F(n,5,2) \quad (n\geq 8)$$

其中 $F_4(1,2)=0$， $F_4(2,2)=1$， $F_4(3,2)=3$， $F_4(4,2)=0$， $F_4(5,2)=20$， $F_4(6,2)=184$， $F_4(7,2)=1344$。 □

(二) $F_4(n,3)$的遞迴關係式

定理 18

$$F_4(n,3)=C_1^n\times F(n-1,5,2)+C_2^n\times F(n-2,5,1)+C_2^n\times F(n-2,5,2)+C_3^n\times F(n-3,5,2)$$
$$+C_3^n\times C_3^{n-3}\times\frac{1}{2!}\times F(n-6,5,1)+F(n,5,3) \quad (n\geq 11)$$，

其中$F_4(1,3)=0$，$F_4(2,3)=0$，$F_4(3,3)=1$，$F_4(4,3)=0$，$F_4(5,3)=0$，$F_4(6,3)=0$，$F_4(7,3)=504$，$F_4(8,3)=8512$，$F_4(9,3)=58400$，$F_4(10,3)=606960$。

證明：

$F_4(n,3)$為 n 個相異元素排列下，符合 4-錯排，且只產生 2 組循環組的方法數。可以分為以下幾種情況，只產生≥ 5的循環組、產生 1~2 個$1-cycle$、產生 1 個$2-cycle$、產生 1~2 個$3-cycle$。只產生≥ 5的循環組，且只產生 2 組循環組即為$F(n,5,3)$。產生 1 個$1-cycle$的情

況，有 n 個數可以做為那一個$1-cycle$，故為C_1^n，而現在要產生 3 個循環組，$F\left(n-1,5,2\right)$表示為$n-1$個相異物進行排列，只產生 2 個循環組，且每組循環組至少 5 個相異物的方法數。產生 2 個$1-cycle$的情況，有 n 個數可以做為那二個$1-cycle$，故為C_2^n，而現在要產生 3 個循環組，$F\left(n-2,5,1\right)$表示為$n-2$個相異物進行排列，只產生 1 個循環組，且每組循環組至少 5 個相異物的方法數。產生 1 個$2-cycle$的情況，產生$2-cycle$的方法數為C_2^n，而現在要產生 3 個循環組，$F\left(n-2,5,2\right)$表示為$n-2$個相異物進行排列，只產生 2 個循環組，且每組循環組至少 5 個相異物的方法數。產生 1 個$3-cycle$的情況，產生$3-cycle$的方法數為C_3^n，而現在要產生 3 個循環組，$F\left(n-3,5,2\right)$表示為$n-3$個相異物進行排列，只產生 2 個循環組，且每組循環組至少 5 個相異物的方法數。產生 2 個$3-cycle$的情況，產生$3-cycle$的方法數為$C_3^n\times C_3^{n-3}\times\frac{1}{2!}$，而現在要產生 3 個循環組，$F\left(n-6,5,1\right)$表示為$n-6$個相異物進行排列，只產生 1 個循環組，且每組循環組至少 5 個相異物的方法數。故$F_4\left(n,3\right)$的遞迴關係式如下：

$$\begin{aligned}F_4\left(n,3\right)=&C_1^n\times F\left(n-1,5,2\right)+C_2^n\times F\left(n-2,5,1\right)+C_2^n\times F\left(n-2,5,2\right)+C_3^n\times F\left(n-3,5,2\right)\\&+C_3^n\times C_3^{n-3}\times\frac{1}{2!}\times F\left(n-6,5,1\right)+F\left(n,5,3\right)\quad\left(n\geq 11\right)\end{aligned}$$

其中 $F_4\left(1,3\right)=0$，$F_4\left(2,3\right)=0$，$F_4\left(3,3\right)=1$，$F_4\left(4,3\right)=0$，$F_4\left(5,3\right)=0$，$F_4\left(6,3\right)=0$，$F_4\left(7,3\right)=504$，$F_4\left(8,3\right)=8512$，$F_4\left(9,3\right)=58400$，$F_4\left(10,3\right)=606960$。 □

六、 *k*-錯排循環組數

當$k=1,2,3$時，我們可以統整出$F_k\left(n,j\right)$的遞迴關係式，如下所示：

$F_1\left(n,j\right)=\left(n-1\right)F_1\left(n-2,j-1\right)+\left(n-1\right)F_1\left(n-1,j\right)$ (引理 7[2])

$F_2\left(n,j\right)=C_1^n\times F\left(n-1,3,j-1\right)+F\left(n,3,j\right)$ (定理 11)

$F_3\left(n,j\right)=C_1^n\times F\left(n-1,4,j-1\right)+C_2^n\times F\left(n-2,4,j-2\right)+G\left(n,3,j\right)$ (定理 16)

此外，我們於推導定理 16 時，得知與 k 互質且小於 k 的$m-cycle$情況與 j 呈現正相關，因此當$k=4$時，$3-cycle$的情況會隨著 j 增加而增加，但目前我們無法採用$G\left(n,4,j\right)$來解決，

因為$G(n,4,j)$會包含產生兩個或兩個以上的$2-cycle$將不符合4-錯排。因此當$k\ge4$時，目前只能分別求出其遞迴關係式。

七、 *k*-錯排演算法

除了遞迴關係式外，我們也嘗試使用C語言透過演算法，進行數值模擬。

(一) $\left|D_{k,n}\right|$的演算法

透過演算法1，本研究以C語言進行數值模擬(程式碼詳見於研究日誌中)，即可求得$\left|D_{k,n}\right|$。

演算法1 ($\left|D_{k,n}\right|$的值)

輸入： $\left|D_{k,n}\right|$的k、n

輸出： $\left|D_{k,n}\right|$

方法：

步驟1 $sum\leftarrow0;num\leftarrow1$

步驟2 $l\leftarrow1$

步驟3 $I_l=\{1,2,3,\cdots,l\}$

步驟4 $\sum_{i\in I}n_i=n$

步驟5 選出$\sum_i n_i\ne k$所有i的組合，如果沒有，則跳至步驟12，否則往下執行

步驟6 記錄此分割有多少相同分區；$x\leftarrow n$。

步驟7 $j\leftarrow$計算n_i 的相同數

步驟8 $num=num\times\left(C_{n_i}^{x}\times(n_i-1)!\right)$

步驟9 $x=x-n_i$

步驟10 若$x=0$往下執行，否則回到步驟8。

步驟11 運算$num=\dfrac{num}{j!}$，$sum=sum+num$，$num=1$，回到步驟5。

步驟12 輸出sum

(二) $F_k(n,j)$演算法

透過演算法 2，本研究以 C 語言進行數值模擬(程式碼詳見於研究日誌中)，即可求得

$F_k(n,j)$。

演算法 2 ($F_k(n,j)$ 的值)

輸入： $F_k(n,j)$ 的 k、n 、j

輸出： $F_k(n,j)$

方法：

步驟 1 $sum \leftarrow 0 ; num \leftarrow 1$

步驟 2 $l \leftarrow 1$

步驟 3 $I_l = \{1, 2, 3, \cdots, l\}$

步驟 4 $\sum_{i \in I}^{`} n_i = n$

步驟 5 選出 $\sum_{i}^{`} n_i \neq k$ 的所有 i 的組合，如果沒有，則跳至步驟 13，否則往下執行

步驟 6 若選出的 n_i 組合中的元素個數 $\neq j$ ，則回到步驟 5。

步驟 7 記錄此分割有多少相同分區； $x \leftarrow n$ 。

步驟 8 $j \leftarrow$ 計算 n_i 的相同數

步驟 9 $num = num \times \left(C_{n_i}^{x} \times (n_i - 1)!\right)$

步驟 10 $x = x - n_i$

步驟 11 若 $x = 0$ 往下執行，否則回到步驟 9。

步驟 12 運算 $num = \dfrac{num}{j!}$ ， $sum = sum + num$ ， $num = 1$，回到步驟 5。

步驟 13 輸出 sum

八、 數值分析

(一) $|D_{k,n}|$ 的值

我們將透過遞迴關係式以及演算法 1 求得的 k-錯排的方法數整理成下表 1。在表 1 中發現其值最大的是 $|D_{5,10}|$ 。除此之外，我們可以觀察到對稱性，也就是 $|D_{k,n}| = |D_{n-k,n}|$ ，這是由於當 n 分割成 k 及 n–k 兩個整數時，若任意 k 個元素所形成的子集，經過簡單排列後，沒有對應到自己本身所成集合，意味著另外 n–k 個元素所形成的子集，經過簡單排列後，也無法對

應到自己本身所成集合。而根據表 2 歸納，我們觀察到當n越大，且k越接近$\frac{n}{2}$時，$\left|D_{k,n}\right|$的值越大。

表 1：$\left|D_{k,n}\right|$之規律性

k \ n	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	1	1	1	1	1	1
2	1	0	2	2	2	2	2	2	2	2
3	2	2	0	6	6	6	6	6	6	6
4	9	14	9	0	24	24	24	24	24	24
5	44	54	54	44	0	120	120	120	120	120
6	265	304	459	304	265	0	720	720	720	720
7	1854	2260	2568	2568	2260	1854	0	5040	5040	5040
8	14833	18108	20145	26704	20145	18108	14833	0	40320	40320
9	133496	161756	176076	200240	200240	176076	161756	133496	0	362880
10	1334961	1618496	1833741	1931616	2492225	1931616	1833741	1618496	1334961	0

我們將上述表格轉換為折線圖(圖 7)，我們可以更清楚的比較，它的成長幅度，由此可驗證當 n 越大，其成長幅度越大，即表示其值越大。

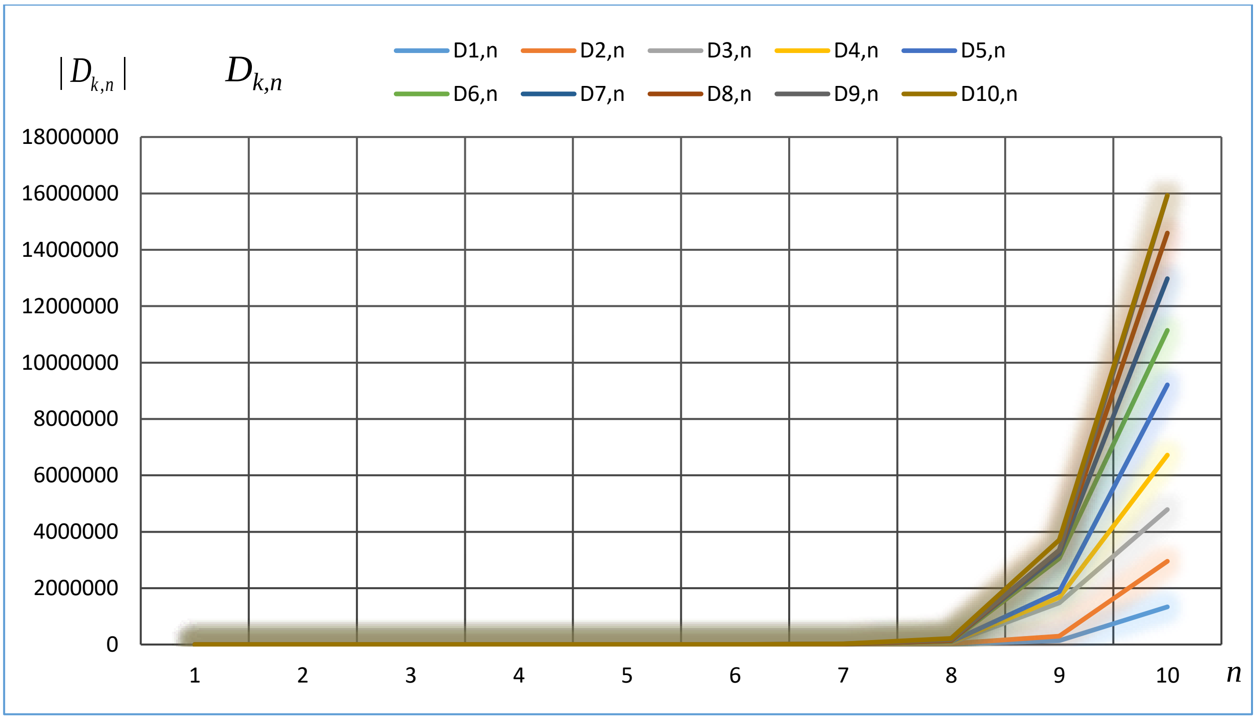

圖 7 $\left|D_{k,n}\right|$趨勢圖

(二) *k*-錯排數值

我們將將前面透過遞迴關係式和演算法 2 求得的$|d_{k,n}|$、$|D_{k,n}|$、$F_k(n,j)$的值整理成表格，如下表 2 所示：

表 2 透過遞迴關係式和演算法 1 所做的數值統整

n	$\lvert D_{1,n}\rvert$	$\lvert d_{1,n}\rvert$	$F_1(n,1)$	$F_1(n,2)$	$F_1(n,3)$	$F_1(n,4)$	$F_1(n,5)$	$F_1(n,j)$
1	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	1
3	2	2	2	0	0	0	0	2
4	9	9	6	3	0	0	0	9
5	44	44	24	20	0	0	0	44
6	265	265	120	130	15	0	0	265
7	1854	1854	720	924	210	0	0	1854
8	14833	14833	5040	7308	2380	105	0	14833
9	133496	133496	40320	64224	26432	2520	0	133496
10	1334961	1334961	362880	623376	303660	44100	945	1334961

n	$\lvert D_{2,n}\rvert$	$\lvert d_{2,n}\rvert$	$F_2(n,1)$	$F_2(n,2)$	$F_2(n,3)$	$F_2(n,4)$	$F_2(n,5)$	$F_2(n,j)$
1	1	0	1	0	0	0	0	1
2	0	0	0	0	0	0	0	0
3	2	2	2	0	0	0	0	2
4	14	6	6	8	0	0	0	14
5	54	24	24	30	0	0	0	54
6	304	160	120	184	0	0	0	304
7	2260	1140	720	1260	280	0	0	2260
8	18108	8988	5040	9708	3360	0	0	18108
9	161756	80864	40320	83664	37772	0	0	161756
10	1618496	809856	362880	799776	433440	22400	0	1618496

n	$\lvert D_{3,n}\rvert$	$\lvert d_{3,n}\rvert$	$F_3(n,1)$	$F_3(n,2)$	$F_3(n,3)$	$F_3(n,4)$	$F_3(n,5)$	$F_3(n,j)$
1	1	0	1	0	0	0	0	1
2	2	1	1	1	0	0	0	2
3	0	0	0	0	0	0	0	0
4	9	9	6	3	0	0	0	9
5	54	24	24	30	0	0	0	54
6	459	225	120	234	105	0	0	459
7	2568	1224	720	1344	504	0	0	2568
8	20145	11025	5040	10380	4620	105	0	20145
9	176076	93456	40320	89424	46332	0	0	176076
10	1833741	965601	362880	853776	540540	75600	945	1833741

n	$\lvert D_{4,n}\rvert$	$\lvert d_{4,n}\rvert$	$F_4(n,1)$	$F_4(n,2)$	$F_4(n,3)$	$F_4(n,4)$	$F_4(n,5)$	$F_4(n,j)$
1	1	0	1	0	0	0	0	1
2	2	1	1	1	0	0	0	2
3	6	2	2	3	1	0	0	6
4	0	0	0	0	0	0	0	0
5	44	44	24	20	0	0	0	44
6	304	160	120	184	0	0	0	304
7	2568	1224	720	1344	504	0	0	2568
8	26704	12208	5040	11808	8512	1344	0	26704
9	200240	88640	40320	91440	58400	10080	0	200240
10	1931616	956016	362880	875376	60690	86400	0	1931616

由上表 2 可發現 $F_k(n,j)$ 同樣具有對稱性 $F_k(n,j)=F_{n-k}(n,j)$，且我們也發現 $\lvert D_{1,n}\rvert=\sum_{j=1}^{n}F_1(n,j)$、$\lvert D_{2,n}\rvert=\sum_{j=1}^{n}F_2(n,j)$、$\lvert D_{4,n}\rvert=\sum_{j=1}^{n}F_4(n,j)$ 及 $\lvert D_{3,n}\rvert=\sum_{j=1}^{n}F_3(n,j)$，故提出定理 19。

定理 19：$\lvert D_{k,n}\rvert=\sum_{j=1}^{n}F_k(n,j) \quad (k\in\mathbb{N})$

證明：

由於 $F_k(n,j)$ 表示為 n 個相異物於符合 k-錯排時產生 j 組循環組的方法數，將其全部加

總，即為 n 個相異物於符合 k-錯排時產生任意循環組數的方法數，即為 n 個相異物進行直線排列符合 k-錯排的排列，即為 $\left|D_{k,n}\right|$。 □

九、 網路爬蟲

(一) 網路爬蟲的遍歷行為

網路爬蟲本質為廣度優先搜索，而亦即遞迴。而網路爬蟲對網際網路進行搜索，設一個網站為一個點，連接兩個網站的網址為一條邊，即表示對一張圖 $G=(V,E)$ 進行搜索，V 為圖 G 中所有點所成集合，E 為圖 G 中所有邊所成集合[3]。也就是典型的 NP-hard 的哈密頓路徑問題。

定義5：若n個相異物進行排列所對應的圖為 $G=(V,E)$，其中 $|V(G)|=n$ 且 $E(G)$ 為此n個相異物進行排列所對應之關係。則 $D_{k,n}$ 可以表示為 D_{k,K_n} 、$F_k(n,j)$ 可表示為 $F_k(K_n,j)$。

而 D_{k,K_n} 就可以表示為當有 n 個網站連接情形為 K_n 時，符合 k-錯排的爬行疆域所形成的集合。$F_k(K_n,j)$ 即可表示為當有 n 個網站連接情形為 K_n 時，共有 j 隻網路爬蟲進行搜索時，符合 k-錯排的爬行疆域所形成的集合。

而網路爬蟲進行遞迴時，須要參數作為指引，往哪裡進行遞迴搜索，效率會比較好或情況會比較多。當網站間連結情形為 K_6 時，如下表 3 所示，網路爬蟲往 k=3 的方向遞迴，其情況數會比 k=1,2,4,5 時來得多。

表 3：網站間連結情形為 K_6

k	1	2	3	4	5
K_6	265	304	459	304	265

而當網路決定往 k=3 的方向搜索時，如下表 4 所示，由 2 隻網路爬蟲負責效率會比由 1 隻或 3 隻來得好，由於數目太少會造成搜索時間太長，數目太多又會互相影響，造成網站負擔。

表 4：網站間連結情形為 K_6

$F_3(K_6,1)$	$F_3(K_6,2)$	$F_3(K_6,3)$	$F_3(K_6,4)$
120	234	105	0

但此參數只能決定搜索方向，不能確定搜索方式。

(二) 最佳爬行疆域

假設網路爬蟲處理量沒有限制時，當 n 個相異物進行直線排列後，任取 $k(k<n)$ 個元素不完全相同時，同時也只產生一個 $cycle$，即表示僅由一隻爬蟲進行運作即可，即為單一網路爬蟲的最佳利用也就是解決哈密頓路徑問題（Hamiltonian path problem），據此提出定理 20，以優化解決廣度優先的網路爬蟲問題。

定理 20：當 n 個相異物進行直線排列後，只產生 1 個 cycle 的情況，所對應的爬行疆域即達到單一網路爬蟲最佳利用。

證明：

排列情況中，對於所有正整數 $k(k<n)$，符合所有 k-錯排的排列情況，在整數分割分類中，只有 n 的情況，因為此分類只會產生一個 n-cycle，而在只有 n-cycle 的情況下，必不符合 n-錯排，且符合 1-錯排~$n-1$-錯排，由此得證。 □

我們設一隻網路爬蟲，從一個網站經過一個網址抵達另一個網站所需的時間為 $t=1$，如下圖 8。若網路爬蟲數目沒有限制時，我們於研究中發現，n 個相異物進行直線排列，不考慮 1-cycle 的情況，因為 1-cycle 即表示網路爬蟲尚未搜索到的區域。而我們發現當 n 為偶數和奇數時，其所需最短時間為 2、3，即定理 21。

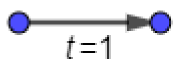

圖 8 移動所需時間

定理 21：當 n 個相異物進行直線排列，n 為奇數時，其最短時間為 3。而 n 為偶數時，其最短時間為 2。

證明：

當 n 為偶數時，兩兩進行交換，即形成 $\frac{n}{2}$ 個 2-$cycle$，而每個 cycle 為為各個獨立的網路爬

蟲進行運作，同時間進行，因此最短時間為 2。當 n 為奇數時，同理兩兩進行交換，但會多出一個 1-*cycle*，但我們不考慮 1-*cycle* 的情況，因此將一個 2-*cycle* 改為 3-*cycle*，最短時間為 3。

□

(三) 網路爬蟲改良說明

我們以定理 20 及 21 的方法修正文獻[10]中的基礎分散網路爬蟲演算法，結果如下：

分散式網路爬蟲演算法改良

Step 1：將整個網路的所有 URL 檢視一次判斷是否為所要的特定集合 A={1,2,3} or B={4,5} or C={ } or …。若判斷為非特定的集合，則不做處置。

Step 2：判斷 Step 1 的特定集合 A or B or C or …是否為空集合：

(a) 若為非空集合(如 STEP1 中的 A，B)，則對此集合內的元素進行**定理 20 及 21 的錯排運算**，以加速滿足運算，找出是否為網路爬蟲所要進行的特定活動。

(b) 若為空集合(如 STEP1 中的 C)，則不做處置。

我們以圖9為例：

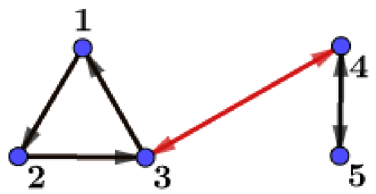

圖9 廣度優先分散式網路爬蟲改良演算法遍歷模擬

首先將將整個網路的所有 URL 檢視一次是否為所要的特定集合 A={1,2,3} or B={4,5} or C={ }。若為非特定的集合，則不做處置。再判斷是否為空集合，如 C 即為空集合，則不做處置。而 A、B 為非空集合進行**定理 20 及 21 的錯排運算**，如圖 9，結點 3、4 先依照定理 20 各自相連後，再依照定理 21 相連，即完成圖 9 的遍歷。

而網站間連接情形更為複雜時，我們則須透過 D_{k,K_n} 、$F_k(K_n, j)$ 參數來決定搜索方向，使得分散式網路爬蟲能依照**定理 20 及 21 錯排運算**進行遍歷。

十、 文獻[13]開放問題

文獻[13]中提出猜想：$\left|D_{k,n}\right| \equiv 0 \pmod k$，促使我觀察我研究中錯排相關數據，並發現相同性質，並提出定理 22、23、24、25、26、27。

定理 22：$F(n,i,j) \equiv 0 \pmod{(i-1)}$

證明：

由引理 4 可得知 $F(n,i,j)=(n-1)F(n-1,i,j)+\left(\prod_{k=1}^{i-1}(n-k)\right)F(n-i,i,j-1)$，其中

$F(n,i,j)=0 \quad (n<i\times j)$，$F(n,i,1)=(n-1)! \quad (n\geq i)$。可得知當 $j=1$ 時，$F(n,i,1)\equiv 0 \pmod{(i-1)}$。

設 $j=x$ 時，$F(n,i,x)\equiv 0 \pmod{(i-1)}$。則當 $j=x+1$ 時，由引理 4 可得知當 $n=1$ 時，

$F(1,i,x+1)\equiv 0 \pmod{(i-1)}$。設 $n=y$ 時，$F(y,i,x+1)\equiv 0 \pmod{(i-1)}$。則當 $n=y+1$ 時，且

由引理 4 得知：$F(y+1,i,x+1)=y\times F(y,i,x+1)+\left(\prod_{k=1}^{i-1}(y+1-k)\right)F(y+1-i,i,x)$，而由前面兩

假設可得知 $F(n,i,x)\equiv 0 \pmod{(i-1)}$、$F(y,i,x+1)\equiv 0 \pmod{(i-1)}$，即得知

$F(y+1,i,x+1)\equiv 0 \pmod{(i-1)}$，得證：$F(n,i,j)\equiv 0 \pmod{(i-1)}$。 □

定理 23：$F_2(n,j)\equiv 0 \pmod 2 \quad (n\geq 2)$

證明：

由定理 11 可得知 $F_2(n,j)=C_1^n\times F(n-1,3,j-1)+F(n,3,j) \quad (n\geq 2)$，且由定理 22 可得知

$F(n,3,j)\equiv 0 \pmod 2$，因此可得知 $F_2(n,j)\equiv 0 \pmod 2 \quad (n>2)$。 □

定理 24：$\left|D_{2,n}\right|\equiv 0 \pmod 2 \quad (n\geq 2)$

證明：

由定理 19 可得知 $\left|D_{2,n}\right|=\sum_{j=1}^{n}F_2(n,j)$，且由定理 23 可得知 $F_2(n,j)\equiv 0 \pmod 2 \quad (n\geq 2)$，

因此可得知 $\left|D_{2,n}\right|\equiv 0 \pmod 2 \quad (n\geq 2)$。 □

定理 25：$G(n,3,j)\equiv 0 \pmod 3 \quad (n\geq 3)$

證明：

由定理 15 可得知

$$\begin{aligned}G(n,3,j)=&(n-1)G(n-2,3,j-1)+(n-1)G(n-1,3,j)-(n-1)(n-2)G(n-3,3,j-1)\\&+(n-1)(n-2)(n-3)G(n-4,3,j-1)\end{aligned},$$

其中 $G(1,3,1)=0$，$G(2,3,1)=1$，$G(3,3,1)=0$，$G(4,2,2)=3$， $G(n,3,j)=0 \quad (n\le 3, j>1)$，$G(n,3,1)=(n-1)! \quad (n>3)$。因此可得知 $G(n,3,1)\equiv 0 \pmod 3 \quad (n\ge 3)$。設 $j=x$ 時，$G(n,3,x)\equiv 0 \pmod 3 \quad (n\ge 3)$。則當 $j=x+1$ 時，由引理 15 可得知當 $n=3$ 時，$G(3,3,x+1)\equiv 0 \pmod 3$。設 $n=y$ 時，$G(y,3,x+1)\equiv 0 \pmod 3$。則當 $n=y+1$ 時，且由引理 15 得知：

$$\begin{aligned}G(y+1,3,x+1)&=y\times G(y-1,3,x)+y\times G(y,3,x+1)-y\times(y-1)G(y-2,3,x)\\&\quad+y\times(y-1)(y-2)G(y-3,3,x)\end{aligned},$$

而由前面兩假設可得知 $G(n,3,x)\equiv 0 \pmod 3 \quad (n\ge 3)$、$G(y,3,x+1)\equiv 0 \pmod 3$，即得知 $G(y+1,3,x+1)\equiv 0 \pmod 3$，得證：$G(n,3,j)\equiv 0 \pmod 3 \quad (n\ge 3)$。 □

定理 26：$F_3(n,j)\equiv 0 \pmod 3 \quad (n\ge 3)$

證明：

由定理 16 可得知 $F_3(n,j)=C_1^n\times F(n-1,4,j-1)+C_2^n\times F(n-2,4,j-2)+G(n,3,j) \quad (n\ge 3)$，且由定理 22 可得知 $F(n,4,j)\equiv 0 \pmod 3$，而由定理 25 可得知 $G(n,3,j)\equiv 0 \pmod 3 \quad (n\ge 3)$，因此可得知 $F_3(n,j)\equiv 0 \pmod 3 \quad (n\ge 3)$。 □

定理 27：$|D_{3,n}|\equiv 0 \pmod 3 \quad (n\ge 3)$

證明：

由定理 19 可得知 $|D_{3,n}|=\sum_{j=1}^{n}F_3(n,j)$，且由定理 26 可得知 $F_3(n,j)\equiv 0 \pmod 3 \quad (n\ge 3)$，因此可得知 $|D_{3,n}|\equiv 0 \pmod 3 \quad (n\ge 3)$。 □

由於尚未推得 $F_k(n,j) \quad (k\ge 4)$ 的遞迴關係式，故未能證明 $|D_{k,n}|\equiv 0 \pmod k$。而我們觀察已推證得的 $F_4(n,j)$ 分項遞迴關係式(定理 17, 18)，確認符合此性質。

參、 研究結果與討論

我們將前面有關 k-錯排性質整理出如下的關係圖：

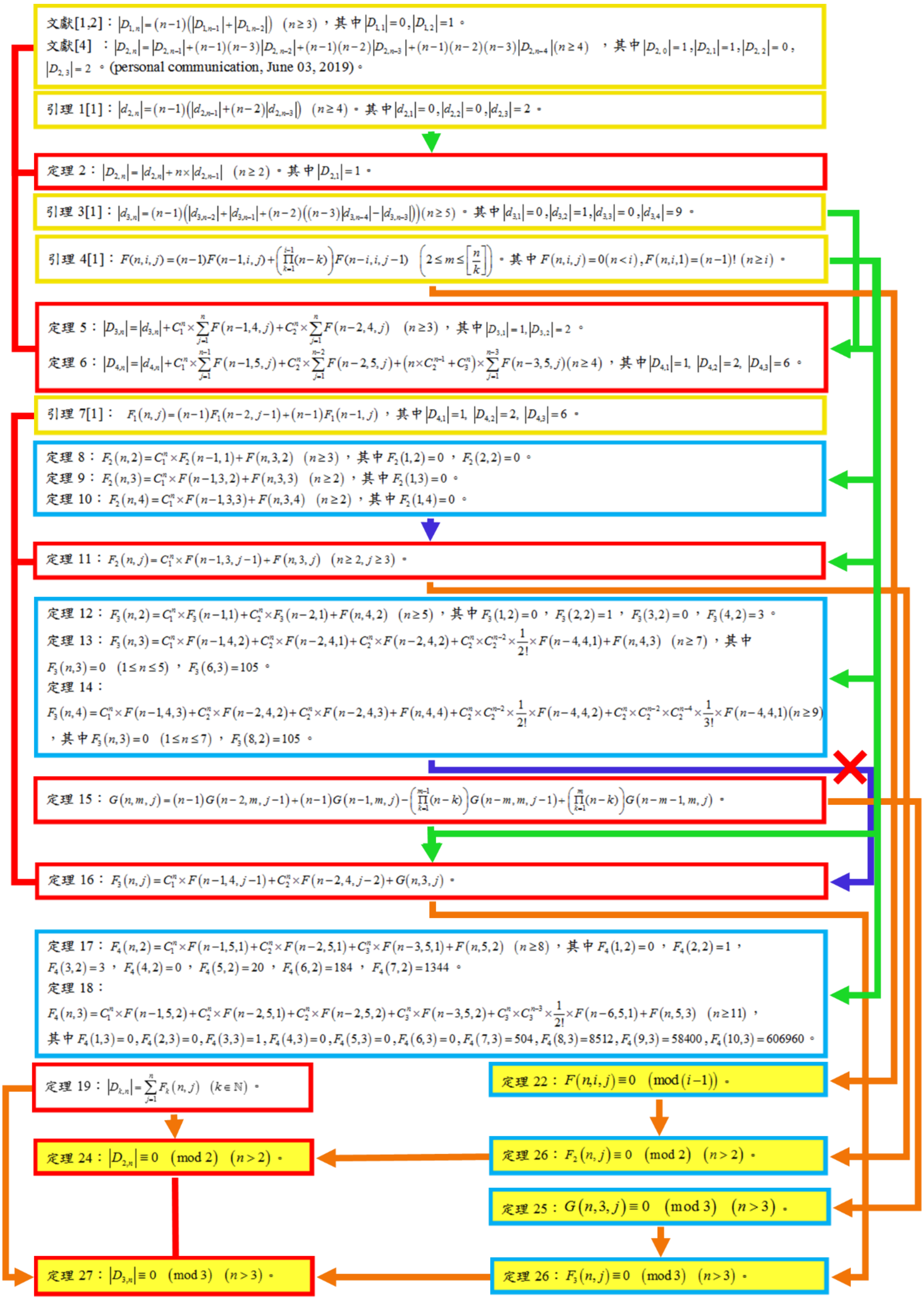

圖 10 k-錯排性質關係圖

目前研究成果曾與美國密蘇里州立大學 Reid L.教授進行討論，教授先前成功透過指數生成函數來推導 $\left|D_{k,n}\right|$ 的遞迴關係式，最後只成功推出 $\left|D_{2,n}\right|$ 的遞迴關係式如下(personal communication, June 03, 2019)：

$$\begin{aligned}\left|D_{2,n}\right| = \left|D_{2,(n-1)}\right| + (n-1)(n-3)\left|D_{2,(n-2)}\right| + (n-1)(n-2)\left|D_{2,(n-3)}\right| \\ + (n-1)(n-2)(n-3)\left|D_{2,(n-4)}\right| \quad (n \geq 4)\end{aligned}$$

其中 $\left|D_{2,0}\right| = 1$，$\left|D_{2,1}\right| = 1$，$\left|D_{2,2}\right| = 0$，$\left|D_{2,3}\right| = 2$。

據教授所言，當 $k \geq 3$ 時，指數生成函數會顯得太過繁瑣而難以推導，而我利用引理 1、3、4 循環組的概念來進行推導，最終成功推導得出下列 $\left|D_{2,n}\right|$、$\left|D_{3,n}\right|$、$\left|D_{4,n}\right|$ 的遞迴關係式。

一、成功推導出 $\left|D_{2,n}\right|$ 的遞迴關係式(定理 2) $\left|D_{2,n}\right| = \left|d_{2,n}\right| + n \times \left|d_{2,n-1}\right| \quad (n \geq 2)$。

二、成功推導出 $\left|D_{3,n}\right|$ 的遞迴關係式(定理 5)

$$\left|D_{3,n}\right| = \left|d_{3,n}\right| + C_1^n \times \sum_{j=1}^{n} F(n-1,4,j) + C_2^n \times \sum_{j=1}^{n} F(n-2,4,j) \quad (n \geq 3)$$ 。

三、成功推導出 $\left|D_{4,n}\right|$ 的遞迴關係式(定理 6)

$$\begin{aligned}\left|D_{4,n}\right| = \left|d_{4,n}\right| + C_1^n \times \sum_{j=1}^{n-1} F(n-1,5,j) + C_2^n \times \sum_{j=1}^{n-2} F(n-2,5,j) + C_3^n \times \sum_{j=1}^{n-3} F(n-3,5,j) \\ + n \times C_2^{n-1} \times \sum_{j=1}^{n-3} F(n-3,5,j) \quad (n \geq 4)\end{aligned}$$

四、成功推導出 $F_2(n,j)$ 的遞迴關係式(定理 11)

$$F_2(n,j) = C_1^n \times F(n-1,3,j-1) + F(n,3,j) \quad (n \geq 2, j \geq 3)$$ 。

五、成功推導出 $F_2(n,j)$ 的遞迴關係式(定理 16)

$$F_3(n,j) = C_1^n \times F(n-1,4,j-1) + C_2^n \times F(n-2,4,j-2) + G(n,3,j) \quad (n, j \geq 3)$$ 。

六、成功推導出 $F_4(n,j) \quad (1 \leq j \leq 3)$ 的遞迴關係式(定理 17、18)，目前只能分別求出其遞迴關係式。

七、提出演算法 1 透過 C 語言模擬 $\left|D_{k,n}\right|$ 的方法數。

八、提出演算法 2 透過 C 語言模擬 $F_k(n,j)$ 的方法數。

九、提出定理 19 得知 $|D_{k,n}|$ 可透過 $F_k(n,j)$ 求得，$|D_{k,n}|=\sum_{j=1}^{n}F_k(n,j) \quad (k\in\mathbb{N})$。

十、 根據 D_{k,K_n}、$F_k(K_n,j)$ 參數及遞迴關係分析網路爬蟲遍歷行為並求得最佳網路爬蟲搜索方向。

十一、提出網路爬蟲最佳爬行疆域(定理 20、21)。

十二、透過定理 20、21 提出廣度優先分散式網路爬蟲演算法改良。

十三、成功證得文獻[13]開放問題 $|D_{2,n}|\equiv 0 \pmod 2 \quad (n\ge 2)$、$|D_{3,n}|\equiv 0 \pmod 3 \quad (n\ge 3)$ (定理 24,27)。

肆、 結論與應用

由於網站常是分層進行設計的，由上而下分別是最上層的是高級功能變數名稱，之後是子功能變數名稱，子功能變數名稱下又有子子功能變數名稱等等。因此，每個子功能變數名稱可能還會擁有多個同級功能變數名稱，而且 URL 之間有相互連結，由此構成一個複雜的網路。由於網路爬蟲有所謂的廣度優先演算法[11]。廣度優先搜尋法，本質是一種圖形(graph)搜索演算法。從圖的某一節點(vertex 或 node)開始走訪，接著走訪此一節點所有相鄰且未拜訪過的節點，由走訪過的節點繼續進行先廣後深的搜尋，它更是圖遍歷問題的一種 NP-hard。以樹(tree)來說即把同一深度(level)的節點走訪完，再繼續向下一個深度搜尋，直到找到目的節點或遍尋全部節點。它是利用佇列(Queue)來處理，通常以迴圈的方式進行暴力法呈現。而此分析方式恰好是，做網路爬蟲過程中，可以推測深度優先演算法本質上是具遞迴性的方式的[11]。

本研究提出了定理 20 及定理 21 的性質，可有效改善深度優先演算法網路爬蟲瀏覽時間及提升資料覆蓋率之最佳化結果，並透過 D_{k,K_n}、$F_k(K_n,j)$ 參數及遞迴關係來達到廣度優先分散式網路爬蟲改良演算法的網路爬蟲搜索方向。由於網路爬蟲是多面向的工具方法組成的資訊學科議題，因此若只關注某一方式的分析話題或利用關聯分析法演算，可能無法更有效地提升原有方法的效能。由於網路中每一個網站皆可視為一個大集合中的若干子集，每個網站中的資料又可視為子集合內的元素。因此有必要利用各種組合數學的方式來對其進行演算改善，避免網路爬蟲陷入 NP-hard 問題的局部最佳化的分析盲點。

伍、　參考文獻

[1] 名祺、林冠伶、吳雨靜、陳世勳(2010)。**居然是「e」**。中華民國第 50 屆全國中小學科學展覽會。

[2] 林泓嶧、李宜蓁、周家暐(2013)。**以數值方法分析錯排情形下之循環組數**。2013 臺灣區國際科學展覽會。

[3] 張鎮華（2017）。**演算法觀點的圖論**。臺北市： 國立台灣大學出版中心。

[4] 單維彰、鄭惟厚主編(2017)。**高級中學數學課本第二冊**。臺北市：三民書局。

[5] Hathout, L.（黃俊瑋、邱珮瑜譯，2009）。**數學偵探物語**。台北市：書泉出版社。

[6] Borkowitz, D. (2005). **The name game: exploring random permutations**, Mathematics Teacher, 98, 196-204 .

[7] H. Jackson, K. Nyman and L. Reid(2013). **Properties of generalized derangement graphs**, Involve, Vol. 6, No. 1, 25-33.

[8] Tucker, A.(2006). **Applied Combinatorics**, 5th ed., NJ: Wiley.

[9] Trupti V. Udapure, Ravindra D. Kale and Rajesh C. Dharmik(2014). **Study of Web Crawler and its Different Types,** Journal of Computer Engineering, Vol 16, No. 1, 1-5.

[10] Wan, Y., & Tong, H. (2008). **URL assignment algorithm of crawler in distributed system based on hash**. In *IEEE international conference on networking, sensing and control, 2008. ICNSC 2008* (pp. 1632–1635).

[11] **淺談網路爬蟲中深度優先演算法和簡單程式碼實現**(2018 年 11 月 06 日)。ITREAD。取自 https://www.itread01.com/content/1541482332.html。

[12] 陳**等人(2019)。**網路爬蟲最佳化之組合分析探討**。論文發表於第三十六屆組合數學與計算理論研討會。國立交通大學。5 月 24 日至 25 日，2019。

[13] Anthony Fraticelli, Les Reid (2009), Generalized Derangements. Retrieved 2020, January 04 from https://people.missouristate.edu/lesreid/reu/2009/.

【評語】010034

這篇文章在討論分散網路爬蟲搜尋網址問題，它的本質是遍歷完所有的頂點且沒有重複經過，即所謂哈密頓路徑問題，是 NP-hard 問題。這篇論文由 k-錯排遞迴之性質來探討分散式網路爬蟲最佳化問題，最後透過電腦模擬及組合數學分析推導。本研究結果相當豐富，作者在各式各樣的條件下寫下遞迴式，研究錯排個數及循環組數的特殊情形。然而其計算效率仍須進一步電腦實證。