

投稿類別：資訊類

篇名：

以 Nonogram 遊戲架構為基礎的密碼形式

作者：

張庭梧。國立台南一中。高二十七班

指導老師：

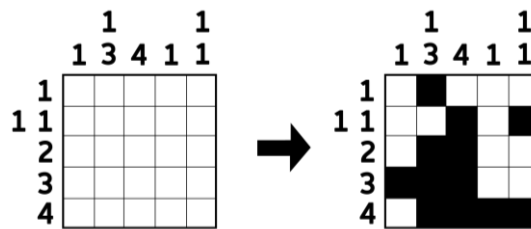
周東賢老師

顏永進老師

壹、前言

一、研究動機

Nonogram，又稱發現小花、數織、Picross，是一種源自日本的邏輯遊戲。玩家必須利用盤面上、左兩側的提示數字作為線索，一步步還原整個盤面上僅由黑白兩種方塊構成的圖形。筆者初次嘗試此種遊戲之後，便喜歡上了這種邏輯推理，進而在遊玩的過程中想到：遊戲的目的是由兩排提示，推導出整個盤面，而如果將英文字母以二進位的黑白色塊表示，再將其拼接起來形成一道 Nonogram 題目。只要將兩排提示傳給接收方，接收方即可透過解開題目，得知圖形，再譯出原本的訊息。這種機制或許可以做為一種密碼。



（圖一：一道 Nonogram 題目與其解答。來源：筆者自行繪製。）

然而在筆者進一步思考這種訊息傳遞方式時，注意到了特殊的現象：同一組提示，可以導出兩種不同但皆合理的圖形。換句話說，使用這種訊息傳遞方式，有可能讓接收者解讀出兩組不同的文句。這樣的現象引起了筆者的好奇，是否能透過在密文中增加輔助機制，防止解密時出現的歧異呢？希望在本文中，能以 Nonogram 為基礎製作出一套可行、無歧異的密碼形式。

二、研究目的

- （一）將 Nonogram 的遊戲模式，應用於資料的加密與解密上。
- （二）設計輔助規則，防止於解密時出現歧異。
- （三）了解在不同變因下，Nonogram 密碼的效率差異。

三、研究方法

- （一）撰寫程式，將加密與解密程序自動化。
- （二）對實際運作後的成果進行數據分析，比較不同變因下 Nonogram 密碼的效率。

貳、正文

一、Nonogram 之遊戲規則

如圖一所示，一道 Nonogram 題目包括兩個要素：長寬不限的矩形網格，和每一直行、橫列皆有的提示數字。每一行或列可以有多個提示數字，合稱為一組，該行或列上每個格子須依照該組提示數字呈現留空（白色）或填滿（黑色）兩種狀態。例如：「1 3 2」代表該行或列上有三條獨立的填滿區段，其長度由上而下或由左而右分別是 1、3、2 格，且每個區塊之間至少有一個格子留空。

玩家可以針對特定行或列上的提示數字進行分析，經由邏輯推論找出一定為黑或白的格子。由於每個格子都是行與列的交錯，解出某一行的格子，等於是幫與該行交集的列提供更多線索，一步一步推理下去即可將整幅圖形解出。在尤瓊雪（2007）研究中，提出了 11 種常見的邏輯判斷捷徑。以下為其中一種：

規則一

一行或列中的格數為 n ，且該行的提示數字只有一個，為 k 。

若 $k > \frac{n}{2}$ ，則該行或列的第 $(n - k)$ 至 $(k - 1)$ 個格子必為填滿的狀態。

（格子的編號由左而右或由上而下為 $0, 1, \dots, (n - 1)$ ）



（圖二：規則一的示例。來源：筆者自行繪製。）

但並不是隨意安排提示數字，都能對應到一張圖形：當推理出矛盾時，即稱此題目無解。此外，有的題目能對應到兩種以上符合其提示數字的答案；有的題目雖然只有單一解，卻無法依靠像上述的純粹邏輯來解決，而需要運用「反覆測試」（trial and error）來破解。例如猜測某一方格為黑色，繼續依邏輯解題下去若出現矛盾，則代表該格子實為白色。出現無解、多重解、猜測解題法等以上情況的題目會被視為不恰當的（nonograms.org, 2019）。

二、將文字轉換為 Nonogram 題目

Nonogram 圖形中，每個格子有留空或填滿兩種狀態，正好可以對應至由 0 和 1 構成的二進位資料。因此，為了將文字以 Nonogram 的形式儲存，必須為英文字母和常見標點符號規定一組對應的編號。顯示如下：

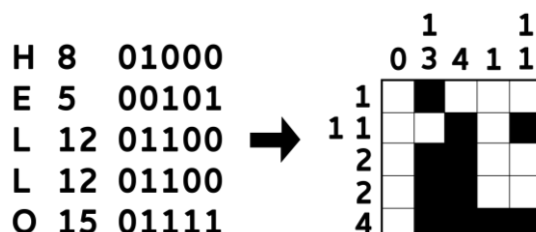
表一：英文字母與英文句號、逗號、空格分別對應的十進位與二進位編號

字元	十進位	二進位	字元	十進位	二進位	字元	十進位	二進位
空格	0	00000	J	10	01010	T	20	10100
A	1	00001	K	11	01011	U	21	10101
B	2	00010	L	12	01100	V	22	10110
C	3	00011	M	13	01101	W	23	10111
D	4	00100	N	14	01110	X	24	11000
E	5	00101	O	15	01111	Y	25	11001
F	6	00110	P	16	10000	Z	26	11010
G	7	00111	Q	17	10001	,	27	11011
H	8	01000	R	18	10010	.	28	11100
I	9	01001	S	19	10011			

為簡化問題，在本研究中不將大小寫分別與數字加入表中。透過以上的對照表，便可以將一長度為 k 的英文字串，轉換為一段長度為 $5k$ 的二進位訊息。例如「HELLO」一字，會

轉換成「01000,00101,01100,01100,01111」。此處於字元之間加入逗號，以利辨認，實際上的轉換並不包含逗號。

下一步是將該二進位訊息轉為只有黑白色塊的 Nonogram 解答圖形，1 代表黑色、0 代表白色，排列的順序是由左到右、由上到下。此處先以格數最接近訊息長度的正方形——五乘五盤面來展示，實際上盤面的長寬是不受限的。構造出圖形後，就可以還原出原始的提示數字。此處的提示數字即是要傳送給接收者的密文。



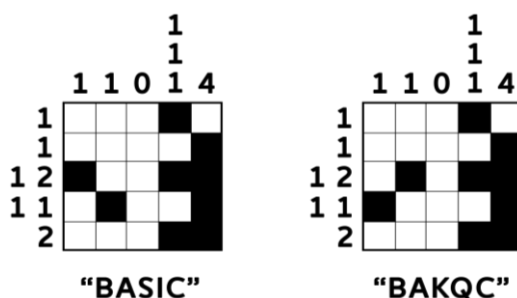
(圖三：「HELLO」一字轉換成的 Nonogram 圖形。來源：筆者自行繪製。)

訊息接收者得到的是一段已轉成字串的提示數字，以圖五為例，「HELLO」一詞可轉為包含逗號的「0,13,4,1,11,1,11,2,2,4」，共 10 組數字。因 10 均分後為 5，前五組數字為直行從左到右的提示，後五組數字則是橫列由上到下的提示，一組代表一排。接收者透過將這十組數字套用在五乘五盤面上，再經邏輯推演得出解答圖形，即可得知原本的二進位資料，進而對照表格解讀訊息。

此外，為方便區別數字（如「112」有可能表示「11,2」、「1,12」抑或「1,1,2」），如出現大於 9 的數字，則以大寫英文字母 A,B,C,...替代。下文將提到的標註碼，也用此方式表達。

然而，解決 Nonogram 已被證明是一個 NP-C 問題（Ueda and Nagao, 1996），意即目前尚未發現能在多項式時間內解決 Nonogram 的演算法。其消耗的時間隨著題目大小增加而成指數成長，若直接將長串文字組合成一個巨大 Nonogram 題目，其解密是耗時且不切實際的，因此訊息的切分為必要。若將二進位訊息拆分成多個固定大小的盤面，即可降低解密難度。例如，一段長 144 位的二進位訊息，可以被拆分成四個六乘六 Nonogram 題目，各自生成一組提示數字。

如上節提到的，一道未經設計的 Nonogram 題目可以有兩種以上的合理解答圖形，因此在字母的特定排列下，最後得到的密文，有可能破譯出兩種以上的明文：



(圖四：具多重解的 Nonogram 題目。來源：筆者自行繪製。)

為了避免歧義，構造更準確的密碼系統，須附加提示數字以外的資訊對解密方向進行補充。在解決 Nonogram 題目的過程中，若該題目具多重解，則透過邏輯推理最終只能確定部分格子的狀態，而留下一區不確定狀態的格子。

例如，若除了提示數字之外，也附上該不確定區域的「哪一格」必為黑色，解密時便有提示可循，最終可還原全部的解答圖形。如圖四中「BASIC」一字，若沒有額外補充資訊，接收者可解得出「BASIC」和「BAKQC」兩種結果。將第(0,2)格（x 軸向右、y 軸向下，(0,0)為最左上格子）標註為必為黑色之後，接收者便能確定只有「BASIC」才是正確的明文。在本研究中，將此種補充資訊稱為「標註碼」。

標註碼的形式為：將被指定為黑色的格子座標(x,y)記為xy，附註在提示數字之後。若有多個須指定的格子，因為一個格子必以兩個字元代表，故不須用逗號分隔。如何決定標註碼提示的格子沒有限制，但數量應愈少愈好（即只提示對解密最關鍵的格子狀態）。

至此，這種密碼可以被描述與定義為：

- （一）基於 Nonogram 遊戲的規則，解密即是解決 Nonogram 題目。
- （二）將字串以 1 與 0 構成的代號轉換。
- （三）密文分為多個固定大小的區塊，各代表一個獨立的題目，將每個區塊各自解出的訊息合併後，即得到原始訊息。
- （四）密文的每一個區塊分為兩個部分，提示數字和標註碼：
 - 1、提示數字必有偶數組，依組數均分成兩段，前半段代表直行的提示數字，後半段則是橫列的提示數字。本研究中，暫只討論正方形的 Nonogram 題目。
 - 2、標註碼指定特定座標的格子必為黑色，以防止解密時出現歧異。

下表為一範例：

表二：「TO BE, OR NOT TO BE.」以三個六乘六盤面表示，紅色代表標註碼顯示之座標

字元編碼	T 10100 O 01111 00000 B 00010 E 00101 , 11011 00000 O 0	O 1111 R 10010 00000 N 01110 O 01111 T 10100 00000 T 10	T 100 O 01111 00000 B 00010 E 00101 . 11100 剩餘 8 格，以 0 填充。
圖形			
提示數字	22,11,21,1,2,1, 11,4,0,11,32,0	11,2,12,12,12,1,2, 5,1,3,4,11,1	21,11,1,0,12,11, 12,2,1,12,2,0
標註碼	無	52	23

三、實作

本研究使用 C++ 語言，撰寫加密器與解密器。首先為使程式容易讀入密文，規定其輸入格式有多行。第一行只有一個數，為盤面的長寬。接著每兩行為一區塊，每區塊中第一行為提示數字、第二行為標註碼，若該區塊不需標註碼，則以英文字母 X 代表。此外，某行或列若沒有任何提示數字，則可省略原本應寫的 0。例如表二的密文，輸入解密器時的形式為：

6
22,11,21,1,2,1,11,4,,11,32,
X
11,2,12,12,121,2,5,1,3,4,11,1
52
21,11,1,,12,11,12,2,1,12,2,
23

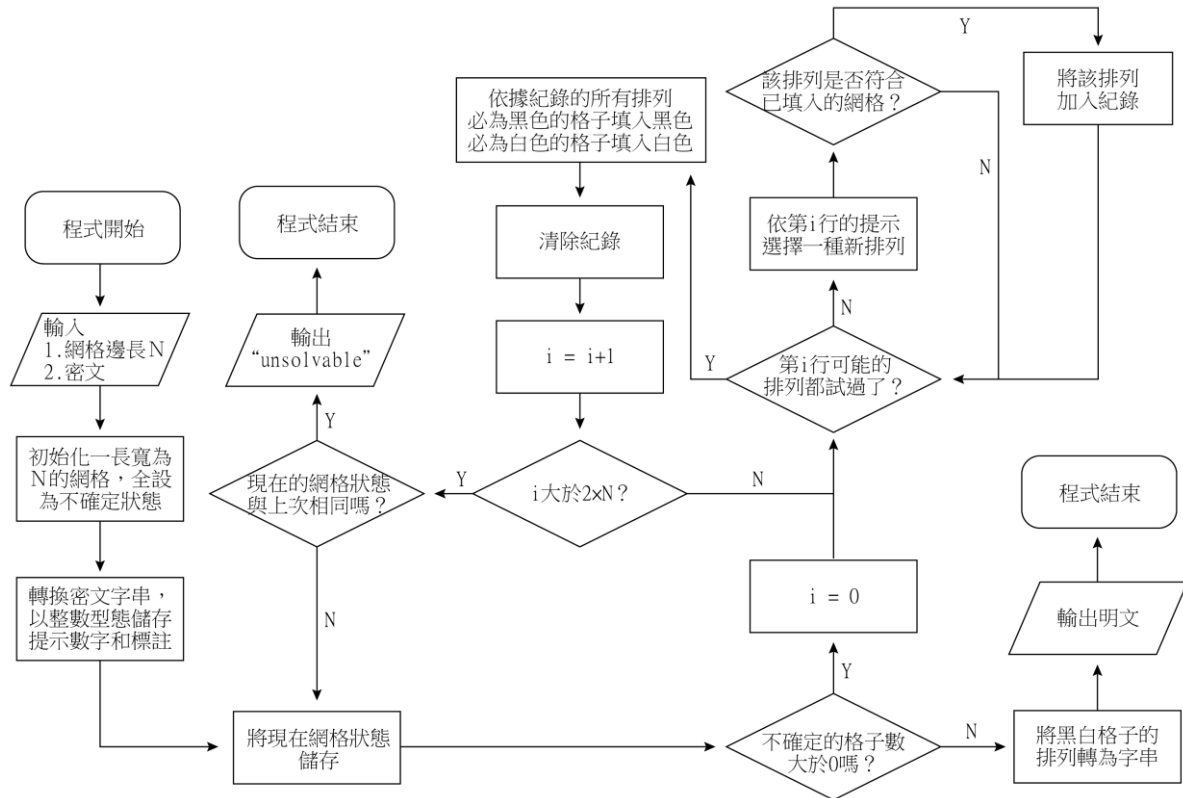
(一) 解密器

解密器在解決 Nonogram 題目，進而轉換成文字訊息時，會先將所有格子狀態設為「不確定」，接著將標註碼代表的座標設為「黑色」。然後程式依序掃過每一行與每一列（在程式中，行與列的編號是連貫的，第一直行的編號為 0、第一橫列的編號即 n ，其中 n 為盤面邊長），對於每一行，遍歷其提示數字有可能代表的所有排列，若該排列不與盤面當前狀況矛盾，則將該排列儲存。遍歷完成之後，若該行有格子於所有儲存過的排列中皆為黑色或皆為白色，即可確定該格顏色，更新當前狀態。一個盤面中，縱橫加總共有 $2n$ 行，這 $2n$ 行經一遍又一遍測試，即可讓不確定狀態的格子數越來越少。當不確定狀態的格子數為 0 時，則題目已經解出，可跳出迴圈開始將黑白格子轉換成文字。

若所輸入的密文不可解或有多重解，則解密器會在解密的過程中不斷遇到矛盾，或線索過少而無法判斷更多的格子狀態。比較兩次掃過 $2n$ 行後，盤面的狀態，若沒有變化，意即沒有得知新的格子狀態，則可斷定程式即使繼續運行下去，也無法進一步解題。此時即可跳出迴圈，回傳「不可解」的結果。

若密文包含多於一個區塊時，則令程式解密完一個區塊後初始化，跳到下一個區塊後重複以上步驟。

表三：Nonogram 密碼解密器之執行流程圖（以單一區塊的解密為例）。



(二) 加密器

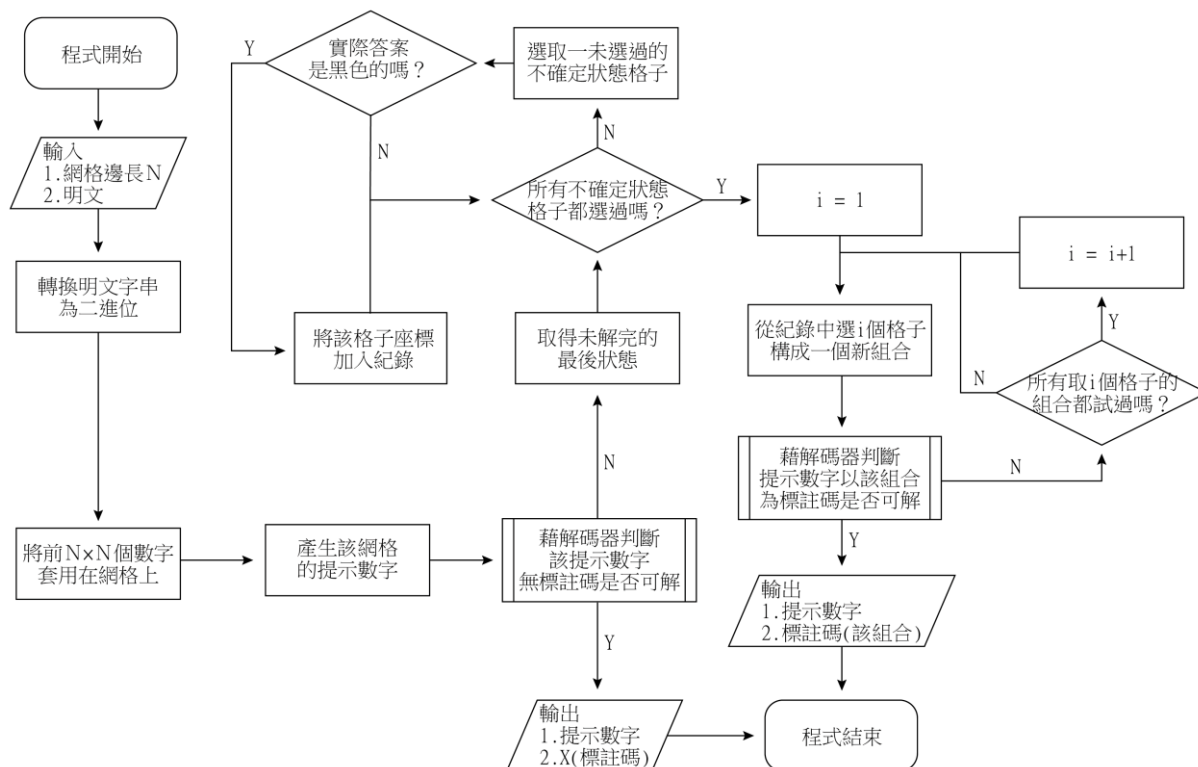
加密器的內部實際上包含了解密器，因為在編排標註碼時，須反覆測試密文是否可解。而本節不再詳述加密器架構中包含的解密步驟。

加密器在接收到一段文字訊息時，會先將其轉為二進位，再依照輸入的盤面邊長 n ，將二進位訊息拆分成數個 n^2 長度的段落，若無法整除，則剩餘欄位補上零，視為空格字元。對於每一個段落，先將二進位訊息填入盤面中，再由產生的圖形，反推出上、左兩邊的提示數字。這一點反而較解決一般 Nonogram 題目（由提示數字推出圖形）簡單，對於長寬為 n 的盤面，只有 $O(n)$ 的複雜度。如果此時單純依靠反推出來的提示數字就可還原圖形，即不須加入標註碼，可以將此區塊的提示數字（也就是密文）直接輸出。

但多數情形並非如此，若此時得出的提示數字經由解密後只能還原部分盤面，而留下許多不確定狀態的格子，即代表這些格子的狀態是解決整個題目的關鍵。下一步是透過遍歷，設計出數量最少的標註碼。首先，程式將比對原始圖形與解密器解密停滯時最終的盤面狀態。將解密器無法推斷其正確狀態，但實際上其狀態為黑色的格子座標儲存——這些格子將作為標註碼的「候選」。先從只有 1 個格子作為標註碼開始測試，從候選名單中取 1 個格子後，讓解密器將其作為標註碼，嘗試繼續解題。若此標註碼使解密器成功解題，則將此標註碼作為正式結果輸出；若不行，則重置盤面，改為取下一個格子為標註碼嘗試解題。若遍歷完從候選名單中取 1 個格子的所有可能後，沒有任何一次能讓解密器成功解題，即開始測試取 2 個格子為標註碼的情況，以此類推。最終得出的標註碼一定是最短的形式。

最後，令程式加密完一個區塊後輸出、初始化，跳到切分的下一段二進位訊息後重複以上步驟即可。

表四：Nonogram 密碼加密器之執行流程圖（以單一區塊的加密為例）



四、變因

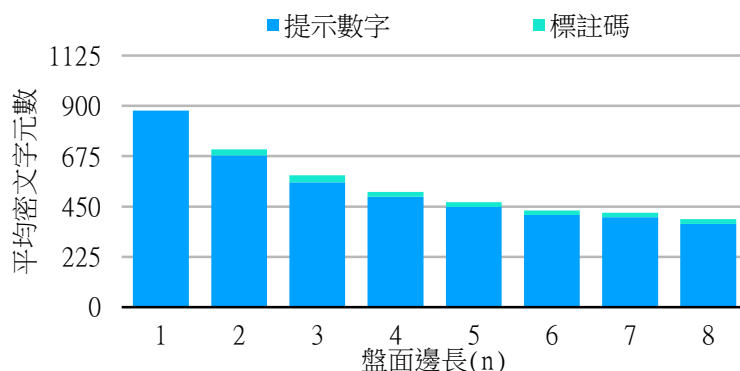
本節將分為兩個方向，探討可能使 Nonogram 密碼效率提升的方法，此處的效率提升意指：將同樣的明文以長度（字元數）更短的密文表示。

（一）改變切分的區塊大小

若將一段長度為 k 的明文，分割成多塊長寬為 n 的盤面，則所有盤面共有 $2n\lceil\frac{5k}{n^2}\rceil$ 組提示數字，每組分別對應一行或一列。隨著 n 的增加，提示數字的組數將減少（如上式，兩者大約成反比）。那麼，密文長度會因而縮短嗎？

此處採用的測試方法為：自英文版維基百科隨機擷取 50 則長度為 100 字元的訊息，分別以不同尺寸的盤面進行加密，再記錄密文中提示數字與標註碼的字元數量（沒有標註碼時輸出的 X 不列入計算）。結果如下表：

表五：加密長度 100 的文字時，在不同盤面邊長下的密文（提示數字加標註碼）平均長度。



表六：自維基百科隨機頁面中挑選的 50 則長度 100 字元的訊息之部分。來源：維基百科。

1	THE HISTORY OF WOMEN FOOTBALL HAS SEEN MAJOR COMPETITIONS...
2	A TEAM SPORT INCLUDES ANY SPORT WHERE INDIVIDUALS ARE...
3	HER FIRST TRADE WAS AS A LONDON-BASED TRANSPORT, WITH...
4	...

由測試結果可知，在 $8 \geq n \geq 1$ 時， n 與密文的長短呈負相關，其中標註碼的長度大多介於 20 至 22 之間，變化不大，可知盤面大小增加後，協助解密的提示不一定會隨之增加。

若繼續嘗試 n 為 9、10 或以上的情況，加密器透過反覆測試選擇標註碼時，所需遍歷之組合數將呈指數上升，使程式的執行時間過長。故在本研究中，暫時只能呈現 $8 \geq n \geq 1$ 的結果，上述趨勢是否延續尚無法確認。

(二) 調整每個字元對應的編號

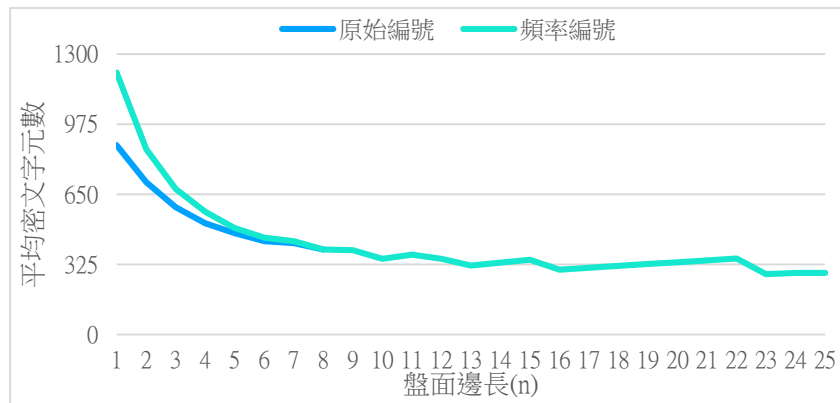
當一個 Nonogram 圖形較為零碎，一行中容易有多個分離的黑色區段，其反推出的每一組提示數字中，出現多個數字的機率也較大，進而增加密文的長度。若依據英文字母的使用頻率多寡，給予使用頻率高的字元較多「1」的二進位編號，轉換成圖形時便可以產生較多黑色格子，降低圖形的零碎程度。這麼做是否能縮短密文長度？首先，將每個字元依據使用頻率，分配新的編號，本研究中將以「頻率編號」代稱之：

表七：依據現代英語的字元使用頻率，重新分配編號，頻率由高而低為左至右、上至下。
數據來源：mdickens.me(2010)、筆者製表

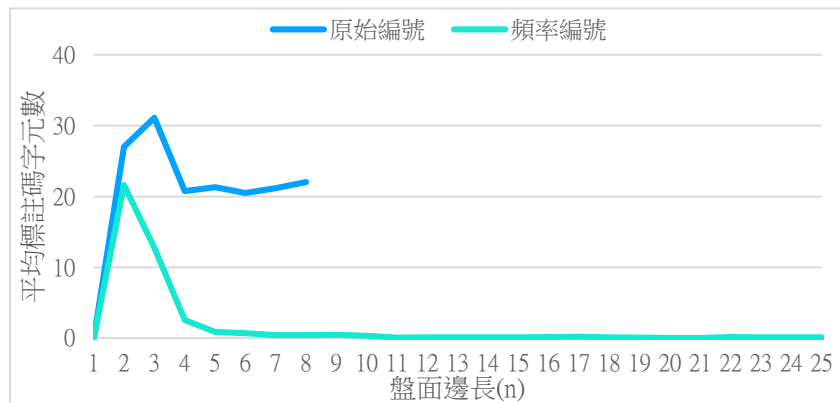
字元	二進位	字元	二進位	字元	二進位
空格	11111	L	11010	B	00101
E	11110	D	10101	,	10100
T	11101	C	01011	.	01001
A	11011	U	10110	V	01010
O	10111	M	01101	K	10000
I	01111	F	01110	X	00001
N	11100	G	11000	J	00010
S	11001	P	10001	Q	00100
R	10011	Y	00011	Z	01000
H	00111	W	10010		

此外，使用原始編號時，若訊息長度未能填滿最後一個區塊，將全部補上 0 作為空格字元，現在使用頻率編號，空格字元應改為填入 1。將同樣的 50 組訊息，用不同盤面邊長 n 進行測試，使用原始編號與頻率編號的加密結果比對如下：

表八：使用原始編號和頻率編號加密長度 100 的文字，在不同盤面邊長下的密文平均長度。



表九：以原始編號和頻率編號加密長度 100 的文字時，不同盤面邊長下的標註碼平均長度。



測試過程中筆者發現，使用頻率編號時的加密效率明顯提升，即使到 $n = 25$ 時仍維持極高的執行效率。且由表九可知，使用頻率編號的平均標註碼長度在 $n \geq 5$ 時，皆介於 0 至 1 之間。故可推測：頻率編號讓黑色格子出現機率增加，該提示數字出現多重解的機率會降低，加密器即不需要額外執行找標註碼的程序，避免了使用原始編號時，執行效率低下的問題。雖然當 $8 \geq n \geq 1$ ，使用原始編號的密文長度皆小於使用頻率編號的密文長度，但執行效率上的優勢讓頻率編號可以以更大的盤面，換取更短的密文，這樣巨大的盤面是使用原始編號時難以實現的。

參、結論

本篇研究利用 Nonogram 遊戲的機制，設計出一種新的密碼形式。這種密碼首先將英文字元替換為 0 與 1 組合的長串訊息，接著將此二進位訊息排列成 Nonogram 的解答圖形。由解答圖形，我們可以反推原始的題目，也就是提示數字，只要將該提示數字傳送給接收者，接收者即可透過解開 Nonogram 題目，得知圖形，進而轉換回原本的英文字串。

然而以上過程中會遇到兩個阻礙：第一，若一次要傳送的訊息太長，形成的 Nonogram 題目會因尺寸過於龐大，解密過程十分耗時而不實際。第二，Nonogram 題目與解答圖形並不總是一對一的關係，在特定情況下，一道題目可以對應到兩種以上的解答圖形，若要直接作為密碼使用，解得的訊息容易出現歧異。

為了解決訊息過長的問題，可將一長串文字拆分成數個小區塊，每個小區塊各自形成一個題目。接收者只須將每一個子題目解出的二進位訊息連接起來，即可得知原本的訊息。這樣的作法成功地使解密加速。

而為了解決解密出現歧異的問題，本研究引入了稱作「標註碼」的概念，與提示數字共同組成密文。標註碼代表的是一或多個座標，被標註的格子必為黑色，而原先該格子在僅有提示數字的時候，是無法被確定狀態的。標註碼為解密提供進一步線索，得以還原正確的圖形。

在上述 Nonogram 密碼的運作方式確立之後，我們以 C++ 撰寫加密與解密的程式，目的在於使這種密碼的使用更為方便快捷，也有助於各種變因的調整、測試的進行。相較於一些論文中嘗試的方法如深度優先搜尋（尤瓊雪，2004）、基因演算法（Wiggers, 2004）。本研究中所採取不斷遍歷可能性的作法較缺乏效率，是未來進一步研究時，首先須改善的工具。

為了讓訊息傳輸的效率更高，本研究中我們也嘗試調整訊息切分尺寸以及每個字母的二進位編號，目的是盡量減少密文的長度。若將每個字元依據使用頻率高低分配編號，使得文字轉換成的圖形中，黑色格子的比例增高。結果顯示，這樣的「頻率編號」可有效減少標註碼的使用，更大幅提升加密器、解密器的運行效率。無論是原始編號或頻率編號，隨著訊息切分尺寸增大，皆顯現出密文縮短的趨勢。雖然在本次測試範圍中，同樣切分尺寸下，使用頻率編號的密文長度皆大於使用原始編號的，但由於運行效率上的優勢，頻率編號在切分尺寸增大時，更加實用。

最後，構造這樣一種 Nonogram 密碼的過程中，產生了許多尚未能回答的問題。第一，程式決定標註碼的方式是不斷嘗試：如果這個格子不行，就換下一個。一直重複到出現正確組合為止。有沒有更準確的數學方法，可以協助分析具多重解的 Nonogram 題目？如此一來便可以設計更簡潔、更有效率的演算法。此外，對於以上測試的結果，密文為何縮短、可能再更簡短嗎？希望在未來研究中，能更深入探討其背後的數學原理。

肆、引註資料

尤瓊雪（2007）。一個有效解決日本益智遊戲「發現小花」的演算法。國立交通大學資訊科學與工程研究所：碩士論文。

Learn to solve Japanese crosswords. Retrieved October 2, 2019, from <https://www.nonograms.org/instructions>

Philosophical Multicore. Retrieved September 28, 2019, from https://mdickens.me/typing/letter_frequency.html

W. A. Wiggers (2004). “A Comparison of a Genetic Algorithm and a Depth First Search Algorithm Applied to Japanese Nonograms,” Twente Student Conference on IT, Jun. 2004.

Nobuhisa Ueda, Tadaaki Nagao (1996). “NP-completeness Results for NONOGRAM via Parsimonious Reductions.” Retrieved October 10, 2019, from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.5277>