The Journal of National Chiao Tung University
Vol. 6, April 1979
pp. 11-22

不定長中文資料之儲存法
An Algorithm of Storing Variables-Length Chinese Data

黃國安・薛榮桃　Kuo-An Hwang and Jung-Tao Hsueh

Department of Computer Science, N. C. T. U.

(Received August 28, 1978)

Abstract — Basing upon the information structure of variable-length Chinese data, this paper will propose and analyze a character-set encoded algorithm for storing them. From this analysis, we find out the condition for which our method will be more efficient than others. Also, by using a sample of one of the most commonly used variable-length Chinese data, the owner's name or title of a typical data processing system, we show that our method does be efficient, economic and practical for storing variable-length Chinese data.

摘要：本文針對不定長中文資料的儲存問題，在一字一碼的假設下，就中文資料的特性，提出字組加碼儲存法，找出其優於傳統方法的條件，並以中文名稱為例，證明其確為一經濟，有效且實用的中文資料儲存法。

一、引　論

近數年來，計算機已成為普遍的工具，其中文化的趨勢，亦日益明朗可行。由於中文字是無法由少數字母拼成的圖形文字，其構造複雜，完全不同於英文字，且所需的資料量更大，故中文資料的儲存問題，為計算機中文化首須解決的項目之一。

在一般資料中，必須以中文形式儲存的各欄（ field ），如姓名、地址、……等，均無固定長度，而不定長的分欄將導至不定長的記錄（ record ），處理時很是困難。通常為了便於處理，不得不把不定長的資料化成定長，甚至不惜以資料項之最大長度來儲存，或截掉過長的資料。前者太浪費空間，且最大之長度常為未知數；而後者却會造成資料不全的遺憾，故此二法，本文不擬再加以討論。

此外，中文資料具有字群（ phrase ）重複出現的特性，譬如在中文名稱中，“股份有限公司”經常出現，如果將這些重複很多的字群，各以一個代碼（ code ）表示，則可節省許多儲存空間。針對具有這種特性之不定長中文資料，本文將提出一種簡便但有效的處理中文資料儲存問題的方法一字組加碼儲存法，並對它加以詳細的分析。

於本文中所論之中文資料，均以一字一碼來儲存，其儲存空間之單位即為一個代碼的長度。原則上每個代碼在 12 數元（ bits ）至 18 數元之間，可依所處理之資料內容及所使用機器之字長（ word-length ）而調整，在此範圍內，約可儲存四千到拾萬字，且適用於大部份的計算機與大部份的中文輸出入系統。

為求敘述的簡潔，下列符號將在本文中被採用：

L：儲存資料所用的定長儲存空間之長度（ Fixed record length ）。

P_i：資料長度為 i 的或然率。

$E := \sum_i iP_i$　資料長度的平均值（期望值，mean ）。

S_n：平均每項資料所需的儲存空間，其下標 n 代表不同的方法。

S_j：字組創新的或然率，其中 j 表字組的字數。

二、傳統之定長儲存法

傳統之定長儲存法為分段儲存法（ Partitioned storing ），其原則為選取一固定是的空間長度 L，以每 L 長的空間為一段，若資料長度不大於 L，則用一段空間儲存之；長度超過 L 的資料，則用若干段空間，到恰好存完為止。

此法所需之平均儲存空間至少為：

$$S_p = \sum_{j=0}^{L} [\ \Sigma\ (j+1)\,L\,P_{jL+i}\ ]$$

$$= E + \sum_{i=1}^{L} [\,(L-i)\ \sum_{j=0} P_{jL+i}\,]$$

$$= E + L - \sum_{j=0}\ \sum_{i=1}^{L} iP_{jL+i} \qquad (1)$$

如果為了存取，異動方便，用一聯繫指標（ linked ）將同屬一項資料的兩段空間聯起來，則需多花聯繫指標的空間做代價。

由式(1)可知，是長儲存法所需的儲存空間，與定長 L 的選取有密切的關係。

三、字組加碼儲存法

此乃針對具有字群重複之特性的不定長中文資料，所設計的儲存方法，目的在利用重複特性，節省儲存空間，且化異長為定長儲存，使資料易於處理。

如果將經常出現的字群，只用一個代碼，稱為多字碼（ multi-character cade ）表示，則可大大地減少儲存空間，但是事先將所有字群搜集編成詞典（ dictionary ）再逐項查詞典，加碼的方法〔1，2〕却非常複雜、麻煩、浪費時間，而且同一資料可能被編成不同的結果，如“股份有限公司”也可能是“（股份）（有限）（公司）”、“（股份）（有限公司）”.....等情況，且結果仍為不定長，因此本文捨棄這種雖然最省空間但耗時且複雜的方法，化繁歸簡，提出此字組加碼儲存法，將不是長的資料，編成固定數目的字組，同時，隨時建立適當的多字碼表，存放所用到的各種多字碼的字群之每個單字，類似詞典，但不必預先編撰，用以維持資料的完整性。

— 茲將此法之編組及加碼法分述於後：

㈠字組編組法：

編組主要以減少創新多字碼為原則，且如上所言，不能增加太多處理的複雜性，因此，如果所選取的固定長度為 L，我們就將資料適當地編成 L 組，其基本法則如下：

(1)若資料之長度不大於 L，則每組最多只有一個字。

(2)若資料之長度大於 L，但不超過 2 L，則每組之字數非一即二。

(3)若資料之長度為 $jL+i$，$j>0$，$0<i\leq L$，則此資料將被編成 i 個（ j+1 ）字的字組及（ $L-i$ ）個 j 字的字組。

顯而易見地，編組方法的選擇與資料的內容有非常密切的關係，例如“中央科學研究院”，若固定長為3，利用不同的編組法，可能編成“（中央）（科學）（研究院）”，“（中央科）（學研）（究院）”，“（中央）（科學研）（究院）”不同的結果，其中以第一種編組結果最好。對於最典型之不是長且必須以中文形式儲存的資料——中文名稱，包括各種法人、公司、機關、行號等，本文提出一種經實驗結果為較理想的編組法如下：

由於中文名稱，出現在後面之字群，重複的可能性最大，如×××公司，而前面之字群也有許多重複的，如台灣××××，所以編組時盡量使後面及前面字組的字數多一點。若固定長選取為L，資料項長度為 jL＋i 時，則將之編成L組，每組原則上為j個字，多餘的i字再一後一前地分配到後前之各組（此處乃指各組的字數分配而言），因此編組後，前後各組有（j＋1）個字，而中間各組別只包括j個字。例如定長為3，則“中央科學研究院”被編為“（中央）（科學）（研究院）”，而“國賓大飯店”則被偏為“（國賓）（大）（飯店）”。

更值得一提的是，本編組法純為計算機內部作業設計的，使用者所接觸的仍為原來完整的資料。

㈡字組加碼法：

編組完成後，分別給每個字組一個代碼，依字組所包括的字數，分為各種多字碼，如兩個字的為2一字碼，k個字的稱為k一字碼，加碼（encoding）必須能區分並表示出各種多字碼，方法很多，各種編碼理論及資訊論〔3，4〕提到很多，故在此不贅述。

㈢編組加碼流程：

當編組法、加碼法及儲存空間之長度L，均選定了後，就可開始進行編組加碼的工作，換舊檔為新檔，其過程如圖一所示，說明如下：

(1)自舊檔依序讀進一項（一字一碼）資料，稱之為A。

(2)若A的長度 |A| ，不大於L，則照原來A之一字一碼存入新檔，到步驟(6)，否則進行步驟(3)。

(3)將A按編組法適當地編成L組，為A_1A_2，……，A_L。

(4)令i從1到L，加碼 A_i，方法如下：

若 $|A_i| \leq 1$，則以原碼 C_i 表示之；否則查尋所對應的 $|A_i|$ 一字碼表，即字數為 $|A_i|$ 之字組的多字碼表，如果找到所要的 A_i，則以 A_i 在多字碼中相對的字碼 Ci 表之，若是找不到 A_i，則將 A_i 加入 $|A_i|$ 一字碼表，產生一新的多字碼 C_i 代表 A_i。

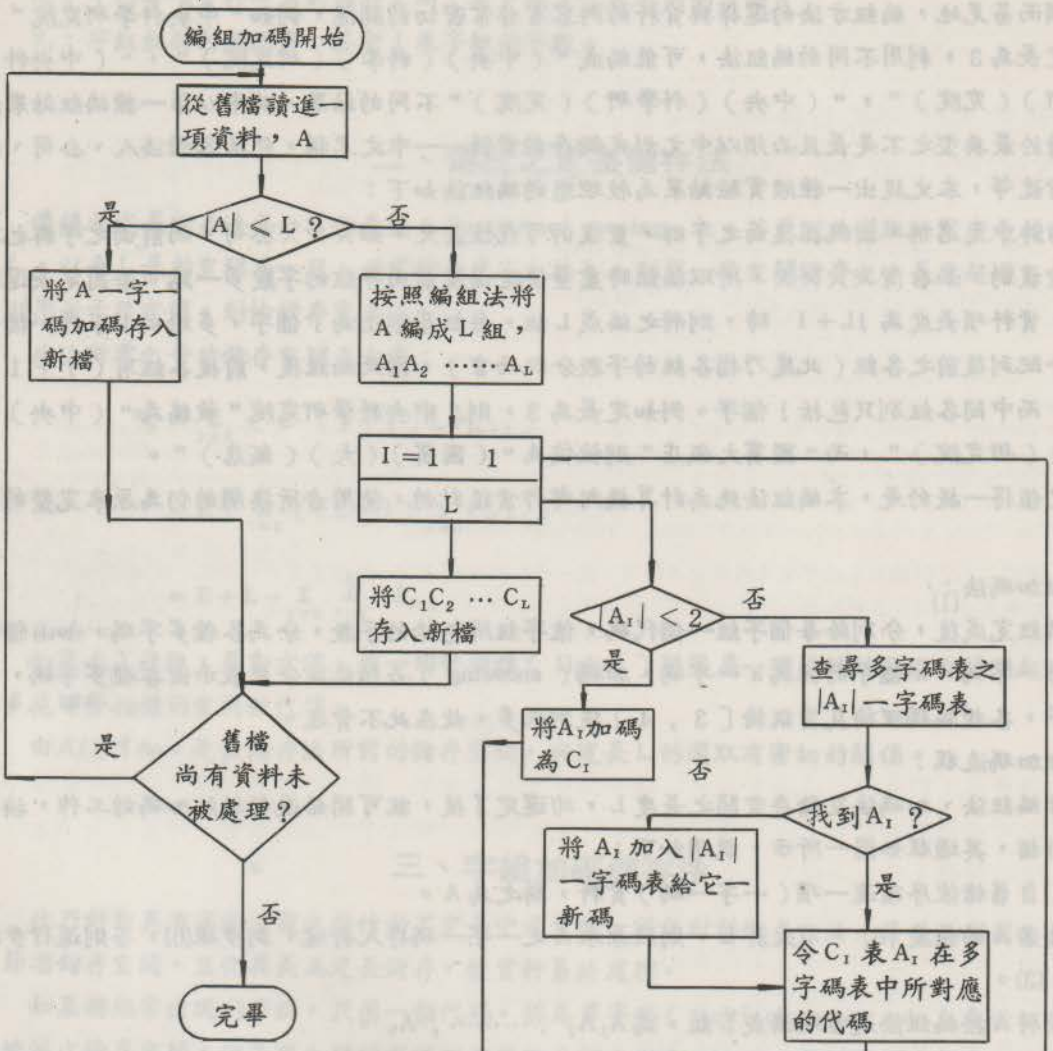
(5)將A加碼完畢成為C_1C_2……C_L後，將之存入新檔。

(6)查看舊檔是否已被處理完畢，否則回到步驟(1)。

(7)結束。

㈣解碼（decoding）：

欲從主檔中取得資料的原始形式，方法很簡單，因為處理過後的新檔所存的是對應字組的代碼，由代碼可分辨出其所代表之多組的種類（即所包括之字數）及其在多字碼表中之位址（address），故可直接由代碼推算出整項資料的長度，並找到其原來字組的內容，因而組合成原始的資料形式。



圖一　字組編組加碼流程圖

四、儲存空間長度的選取

字組加碼儲存法，最重要的項目之一，為儲存空間長度的選取，此與字組創新之或然率有關，也直接影響到全部所需的儲存空間。

於本節先討論字組加碼儲存法所需的儲存空間，再研究理想的固定長度之選取法。

㈠字組加碼儲存法所需之儲存空間：

由編組與加碼法得知，若儲存之固定長度為L時，長度不大於L的資料項僅需L個儲存空間；若資料項長度為L+i，$0 < i \leq L$ 時，則將被編成i組兩個字的字組，假設2一字碼創新之或然率為ρ_2，則平均此項資料需空間$L+2i\rho_2$。同理，當資料項之長度為 $jL+i$，$j \geq 2$，$0 < i \leq L$ 時，此資料將被編成i個（j+1）字的字組與（L－i）個j個字的字組，若 ρ_j 表j一

字碼創新之或然率，則此長度的資料平均需空間$L+i(j+1)\rho_{j+1}+(L-i)j\rho_j$。

故字組加碼儲存法平均所需之儲存空間為：

$$S(L)=\sum_{i=1}^{L} LP_i+\sum_{i=1}^{L}(L+2i\rho_2)P_{L+1}+\sum_{j=2}\sum_{i=1}^{L}[L+i(j+1)\rho_{j+1}$$

$$+(L-i)j\rho_j]P_{jL+1}$$

$$=L+\sum_{j=2} j\rho_j \sum_{i=1}^{L}[iP_{(j-1)L+1}+(L-i)P_{jL+1}] \tag{2}$$

$$=L+E+\sum_{i=1}^{L}\sum_{j=2}[jL(\rho_j-1)+i(\rho_{j+1}-1)+ij(\rho_{j+1}-\rho_j)]P_{jL+1}$$

$$+\sum_{i=1}^{L}[(2i\rho_2-L-i)P_{L+1}-iP_i] \tag{3}$$

由(2)式得知，ρ_j 值越小時，S(L)越小，也就是越省空間。

因為$0\leq\rho_j\leq 1$，$\forall j$，故在最壞的情況下，即所編成的均為新的字組，$\rho_j=1,\forall j$，則此時所需之儲存空間，由（4，3）可得為

$$S_w(L)=L+E+\sum_{i=1}^{L}[(i-L)P_{L+1}-iP_i] \tag{4}$$

$$\geq S(L) \qquad \forall L$$

由(2)式可知，儲存空間S與L有極密切的關係，S隨著L而改變其值。

㈡如何選取儲存的固定長度：

儲存之固定長度的選取，以使儲存空間S(L)越小越好。由於S(L)與L及ρ_j有密切的關係，而其中ρ_j乃代表資料內容不均勻之特性，和編組法有關，亦隨著L而改變，較難估計出其正確值，因此，本文僅考慮最壞的情況，即令$\rho_j=1,\forall j$，以求取最壞情況下最理想的儲存之固定長度。

欲求S_w之最小點（minimum），先考慮其差分方程式（difference equation）

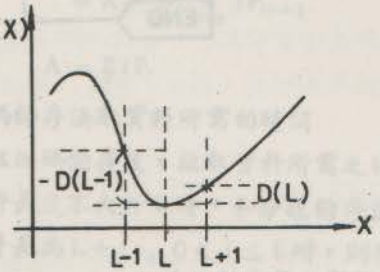
$$D(L)=S_w(L+1)-S_w(L)$$

$$=1-P_{2L+1}-2\sum_{i=1}^{L}P_{L+1} \tag{5}$$

$$=1+2F_{2L}-(F_{2L}+F_{2L+1}) \tag{6}$$

其中$F_L=\sum_{i=0}^{L}P_i$為資料長度的累加機率函數（cumulative distribution function）。

如圖二所示，當$D(L)>0$時，S_w在L點為遞增（increase），



圖二　S_w為極小點的條件

若 $D(L) \leq 0$，則 S_w 在 L 處為遞減（ decrease ），故，若 L 為 S_w 之極小點，則必須滿足

$$D(L-1) \leq 0$$
$$D(L) > 0 \qquad (7)$$

亦即

$$1 + 2F_{L-1} \leq F_{2L-2} + F_{2L-1}$$
$$1 + 2F_L > F_{2L} + F_{2L+1} \qquad (8)$$

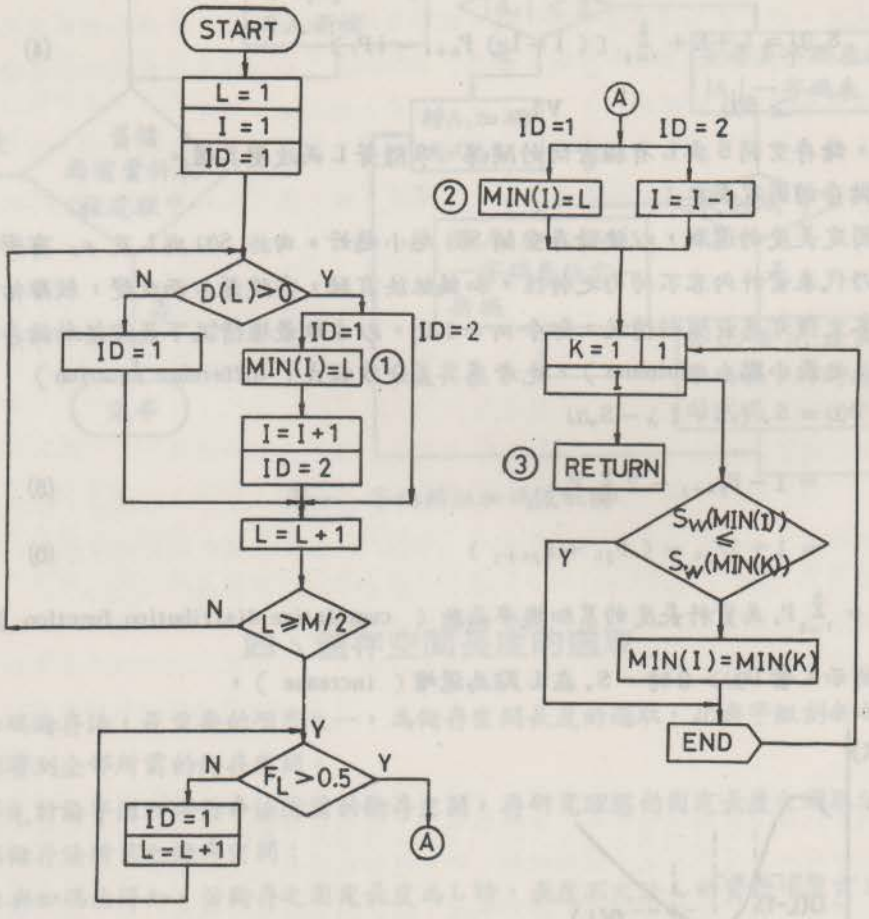
假設 $F_M = 1$ 且 $F_{M-1} < 1$，則當 $L \geq M/2$ 時，$F_{2L} = F_{2L+1} = 1$

則 $D(L) = 1 + 2F_L - 2$

$= 2F_L - 1 \quad > 0 \quad \Rightarrow F_L > 0.5$

此時判別 D(L) 是否大於 0，即為判別 F_L 是否大於 0.5。

綜上所述，得到求 S_w 之最小點之流程，如圖三所示。其中 D，F，S_w 已如上所述，L 代表長度，為 D，F，S_w 之變數，M 為 L 之上限，即最大之長度，I 表極小點之個數，MIN(I) 為存放 I 個極小點之位址，K 為 MIN 之指標，ID 為 1 時表 S_w 在 L－1 為遞減，其他情形 ID＝2



圖三　選取儲存空間長度流程圖（最壞情況）

於圖三中，在③結束時，MIN(1)即為S之最壞情況下的最小點，我們稱之為最小之最差點（min-worst point），即為所求的固定長度。

最小之最差點很容易就可求得，以此當做儲存的固定長度，經實驗後，證明頗為理想。

綜上所述，可知字組加碼儲存法之基本過程如下：

㈠求得資料項長度之機率分佈情形（ probability distribution ）。

㈡觀察資料內容（由抽樣統計或經驗累積獲得），依其特性，決定字組編組法及加碼法。

㈢選取儲存之固定長度。

㈣編組加碼處理。

㈤大功告成。

五、字組加碼儲存法與分段儲存法的比較

字組加碼儲存法與傳統之分段儲存法，同樣都可將不定長之資料，化成定長儲存，但對所需的儲存空間及存取時間而言，則有不同的績效，茲分別比較如下：

㈠儲存空間的比較：

由比較式(1)與(3)可得，就儲存空間而言，字組加碼儲存法優於分段儲存法的條件為 $S(L) < S_p$

即
$$\sum_{i=1}^{L}\sum_{j=2}\left[jL(\rho_j-1)+i\rho_{j+1}+ij(\rho_{j+1}-\rho_j)\right]P_{jL+i}+\sum_{i=1}^{L}(2i\rho_2-L)P_{L+i}<0 \qquad (9)$$

當ρ_1值夠小時，(9)式通常可以成立，也就是說只要字群重複的特性顯著，字組創新的或然率夠小時，字組加碼儲存法就比分段儲存法節省空間。

㈡資料取出之時間的比較：

欲從檔案取出一項完整的原始資料，若該資料之長度大於儲存空間的固定長，則字組加碼法與分段儲存法均需經由聯繫（ link ）之指示，才能達到目的。

本節仍在一字一碼的假設下，討論從主檔案將資料完全解碼，以得到完整的原始資料時，所需參照（ reference ）記憶（ memory ）的次數之比較。

1 分段儲存法取資料所需的時間

利用分段儲存法儲存後，若欲取出長度為 $jL+i$ ，$j>0$ ，$0<i\leq L$ 則必須經過 j 次聯繫，加上（$jL+i$）次一字一碼的解碼，總共需參照記憶（$jL+i+j$）次，因此，平均取一項資料所需參照記憶之次數為

$$T_1(L)=\sum_{i=1}^{L}\sum_{j=0}(jL+i+j)P_{jL+i}$$

$$=A+\sum_{i=1}^{L}\sum_{j=1}jP_{jL+i} \qquad (10)$$

此處 $A=\sum_i iP_i$

2 字組加碼儲存法取資料所需的時間

採用字組加碼儲存後，欲取資料所需之時間分下列情況討論：

①當資料長度不大於 L 時，和分段儲存法一樣，僅需一字一碼之解碼時間。

②若資料長為 $L+i$ ，$0<i\leq L$ 時，則存於檔案中有 i 個 2 一字碼，需 i 次取 2 一字碼表之時間，再加上（$L+i$）個一字碼解碼時間，故總共需參照記憶（$L+i$）$+i$ 次。

③若資料長度為 $jL+i$ ，$j>1$ ，$0<i\leq L$ ，此時檔案中之該資料均為多字碼，故需 L 次取多字碼的時間，加上 $jL+i$ 次一字碼解碼時間，共需參照記憶（ $jL+i+L$ ）次。

因此，此法取資料所需參照記憶之平均次數為：

$$T_2(L)=\sum_{i=1}^{L}\left[iP_i+(L+2i)P_{L+1}+\sum_{j=2}(jL+i+L)P_{jL+i}\right].$$

$$=A+\sum_{i=1}^{L}\left[iP_{L+1}+L\sum_{j=2}P_{jL+i}\right] \tag{11}$$

綜上觀知，當 L 很大時，分段儲存法聯繫的次數很少，而字組加碼儲存法也幾乎全是一字碼，在此情況下，此二法幾無分別。

比較(10)及(11)式，得知字組加碼儲存法優於分段儲存法之條件，就取資料之時間而言，為 $T_2<T_1$

即 $$\sum_{i=1}^{L}\left[iP_{L+1}+L\sum_{j=2}P_{jL+i}\right]<\sum_{i=1}^{L}\sum_{j=1}jP_{jL+i} \tag{12}$$

亦可化成 $$L<\sum_{i=1}^{L}\left[\sum_{j=1}jP_{jL+i}-iP_{L+1}\right]/\sum_{i=1}^{L}\sum_{j=2}P_{jL+i}$$

$$L<\sum_{i=1}^{L}\left[\sum_{j=1}jP_{jL+i}-iP_{L+i}\right]/\sum_{i=2L+1}P_i \tag{13}$$

由(12)或(13)式得悉，若大部份資料的長度大於某一限度，當 L 夠小時，則容易滿足(12)或(13)式之條件。

若將字組加碼儲存法加以修改，則對資料取出之時間有很大的改善，方法如下：

3.對資料取出之時間而言，字組加碼儲存法之改進：

改進的方法為，將多字碼表中所存之一字一碼，直接改換成每個字的實際資訊（ information ），如此一來，每個字組只需參照記憶一次，就可完全得到其資料，節省很多解碼時間，如果字群重複的特性很強，如此修改所增加的儲存空間不會太多。

改進後之字組加碼儲存法，取出一項資料平均所需參照之次數為：

$$T_3(L)=\sum_{i=1}^{L}\left[iP_i+L\sum_{j=1}P_{jL+i}\right]$$

$$=L-\sum_{L=1}^{L}(L-i)P_i \tag{14}$$

由(14)式知，$T_3(L)\leq L$ ，故 L 越小時，速度越快。

㈢資料存放的時間之比較：

利用分段儲存法，存放資料非常簡便，只要將之分成適當的數段，彼此聯繫起來存入即可；但用字組加碼儲存法時，比較麻煩，必須浪費編組及加碼的時間，因此，就這一點比較，分段儲存法較優。

六、實驗結果

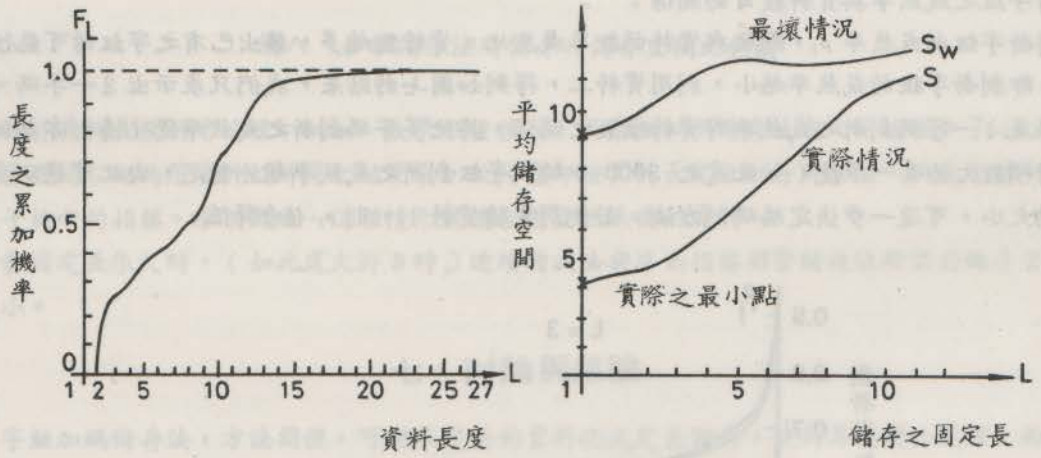
為了使前節所述更具體化，我們選取了 1975 年新竹區汽車的車主名稱作為實驗資料，簡稱為資料一，此資料包括人名、機關、公司、行號之名稱，更有許多資料內容完全相同（包括同名

同姓或一人擁有數車）者。為使實驗結果能接近其他系統，我們又把資料內容完全相同者歸納為一項資料而得資料二，同時進一步除去資料長度為三成以下（大部分是人名）者得資料三用以代表機關、公司、行號之名稱。

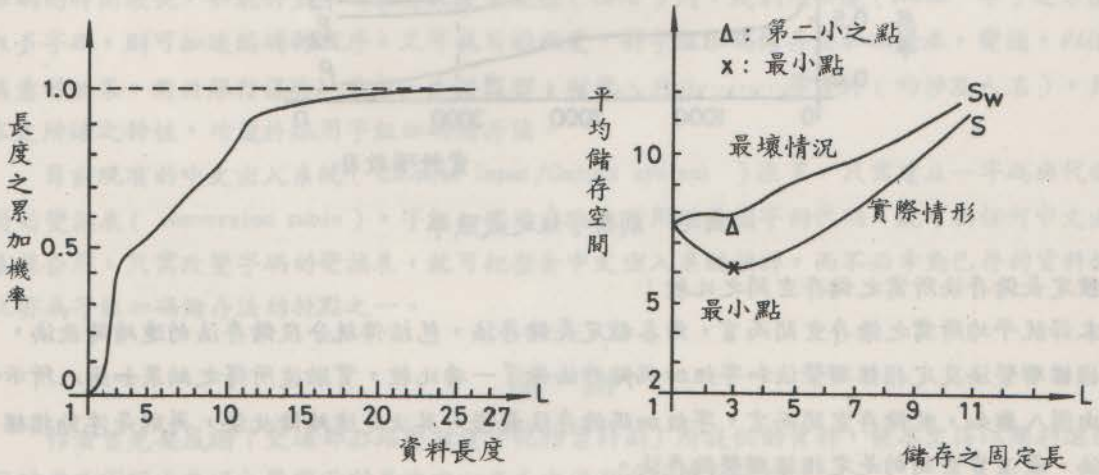
針對上述三種不同的資料，我們各採取至少二十組不同的樣本，分別作實驗，歸納所得之具代表性者說明如下：

㈠儲存空間長度的選取

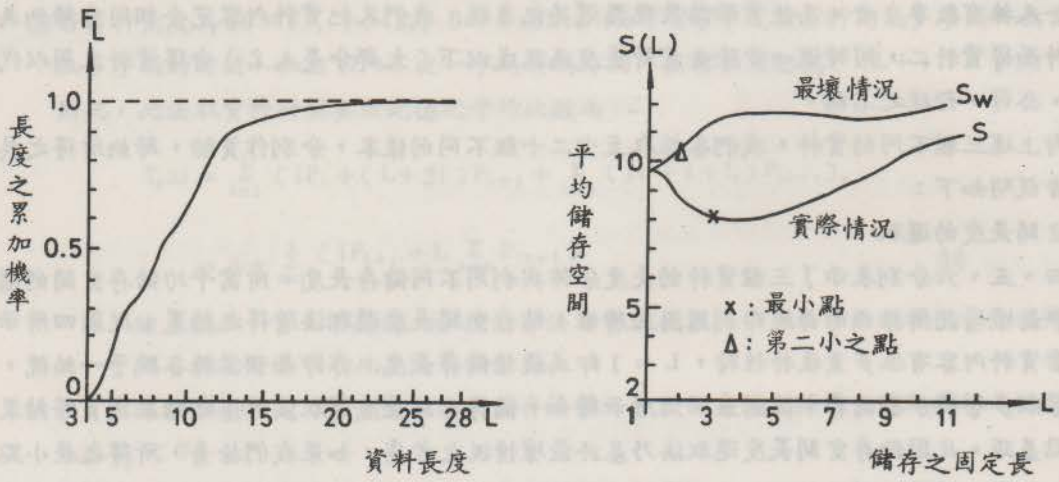
圖四、五、六分別表示了三種資料的長度分佈與利用不同儲存長度，所需平均儲存空間的關係。其中最壞情況所標示的曲綫即利用圖三所示，儲存空間長度選取法所得之結果，從圖四所示得知，當資料內容有很多重複特性時，L = 1 即為最佳儲存長度，亦即每項資料各賦予一編號，資料內容以多字碼分別儲存，從圖五、六所示得知，儲存空間長度選取法所得之結果與實際結果有一點點差距，此因儲存空間長度選取法乃基於最壞情況之考慮，如果我們捨棄，所得之最小點而改用次小點（因最小點為L = 1，而資料重複性為零），則差距就大大地減小，因此所得之效果也算良好，更因儲存空間選取法，只考慮最壞情況，問題簡化許多，故前節所述之選取法，應屬簡易可行之所效方法。



圖四　資料一長度分佈與平均儲存空間圖



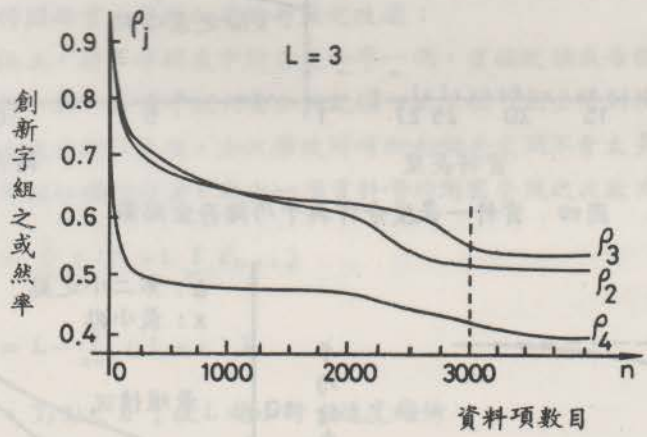
圖五　資料二長度分佈與所需平均儲存空間圖



圖六　資料三長度分佈與所需平均儲存空間圖

㈡創新字組之或然率與資料數目的關係：

創新字組的或然率 ρ_j，顯然與資料的數目成反比，資料數越多，編出已有之字組的可能性就越大，即創新字組的或然率越小，利用資料二，得到如圖七的結果，我們只表示出 2一字碼、3一字碼及 4一字碼創新之或然率對資料項數之關係，其他多字碼創新之或然率變化情形亦類似。當資料項數大於某一限度（如此處之 3000 ）時，字組創新之或然率趨於穩定，由此可臆測多字碼表的大小，可進一步決定編碼的方法。若使用實驗資料 1，則 ρ_j 值會降低。



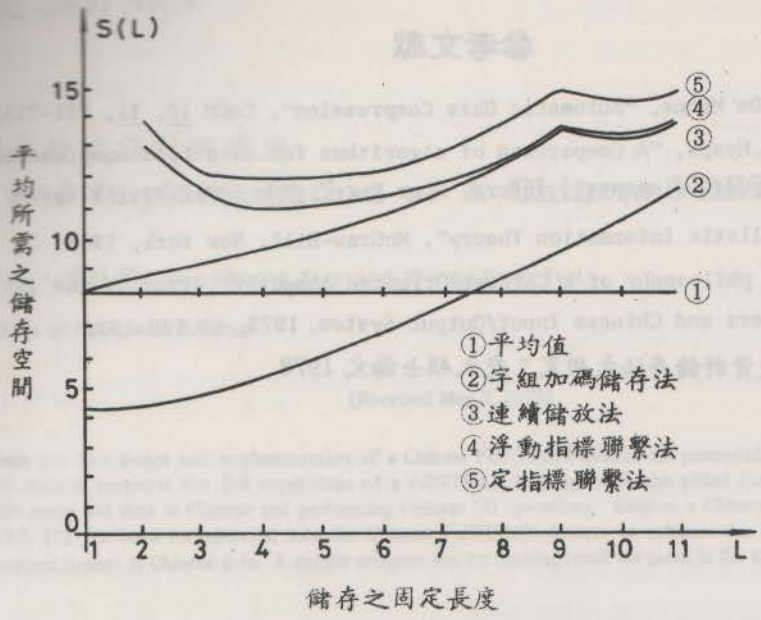
圖七　創新字組之或然率

㈢各種定長儲存法所需之儲存空間之比較：

本節就平均所需之儲存空間而言，對各種定長儲存法，包括傳統分段儲存法的連續儲放法，浮動指標聯繫法及定指標聯繫法和字組加碼儲存法做了一番比較，實驗後所得之結果如圖八所示。

由圖八觀知，就儲存空間而言，字組加碼儲存法最優，其次是連續儲放法，再則是浮動指標聯繫法，最浪費空間的是定指標聯繫儲存法。

在固定長為 1 時，浮動指標聯繫法與定指標聯繫法沒有意義，連續儲放法所需之平均儲存空



圖八　各種儲存法所需平均儲存空間比較圖

間恰等於資料長度的平均值（ mean 數學期望值），而字組加碼儲存法却比平均值少了 46.65％的儲存空間（此為用資料二所得之結果）。此時亦即按資料長度分別存於子檔，而在主檔中僅存其在子檔中的指標。很明顯地，這種情形最適合重複性很高的資料。

當固定長很大時，（如此處大於 8 時）連續儲放法與浮動指標聯繫儲放法所需的儲存空間差異很小。

七、討論與結論

字組加碼儲存法，方法簡便，可將不定長的資料化成定長儲存，使計算機便於處理，而且利用了字群重複的特性，得到節省空間的效果，空間減少了，可處理的資料量就可增大。其缺點為加碼的時間較長，如能將多字碼表存放在主記憶（ core ）內，或利用快速（ hash 等）之方法存取多字碼，則可加速編碼的程序。又可視目的而定，將字組加碼儲存法加以修正，變通，以達到滿意的效果。對於銀行保險、電信、公路監理、財稅、戶政……等資料（均涉及人名），具有本文所述之特性，均適於採用字組加碼儲存法。

目前現有的中文出入系統（ Chinese Input/Output system ）很多，只需建立一字碼與代碼之間的變換表（ conversion table ），字組加碼儲存法中所用的每個字的代碼，就可與任何中文出入系統合用，只需改變字碼的變換表，就可把整套中文出入系統換掉，而不必牽動已存的資料檔，此亦為字組加碼儲存法的特點之一。

誌　謝

作者首先要感謝「交通部公路整理電子化作業計劃」所提供的資料，使本文得以順利進行，同時要感謝國立交通大學電子計算機中心在本文進行中給予的協助。

參考文獻

1. B.A.Marron & P.A.D.De Maine, "Automatic Data Compression", CACM 10, 11, 711-715, (1967)
2. E.J.Schuegraf & H.S.Heaps, "A Comparison of algorithms for data base compression by use of fragments as language elements", Inform. Stor.Retr., 10, 309-319, (1974)
3. Jelinek.F, "Probabilistic Information Theory", McGraw-Hill, New York, 1968
4. John Y.Hsu, "Design philosophy of a Chinese-Oriented computer", Proc of the 1st Int'l Symposium on Computers and Chinese Input/Output System, 1973, 99.139-152.
5. 薛榮桃，“異長中文資料儲存法之研究”交大碩士論文 1978.