接近最佳不定長度錯誤更正碼的建構特性和標準

吳峰蒼 [a]、陳彥銘 [b]
Feng-Tsang Wu [a]、Yen-Ming Chen [b]
空軍航空技術學院航空電子工程科 [a]
國立中山大學通訊工程研究所 [b]
[a] Department of Aero-Electronic Engineering, Air Force Institute of Technology.
[b] Institute of Communications Engineering, National Sun Yat-sen University.

摘要

實際的應用實例及文獻表示中,「聯合訊號源及通道編碼」的方式是優於「分離訊號源及通道編碼」,尤其是對於具有硬體複雜度限制的方案更是如此。在此論文中，提出了能夠為大型來源字母符號構建接近最佳的不定長度錯誤更正碼（VLEC碼）的演算法，非常適用於大型來源字母符號，並且可以避免在建構每個碼字期間產生隨機樣本數，這演算法是根據隨機搜索過程設計出一種具有低搜索複雜度且有效率的演算法，經過使用白色高斯雜訊通道實驗模擬，去評估26個英文字母符號性能，的確在我們建構出具有最小平均碼字長度的VLEC碼簿上，可以呈現保持或改善其錯誤更正之能力，所以，我們所呈現的方案提供了最小平均碼字長度、符號錯誤率性能的改進、降低了演算法搜索複雜性(平均搜索時間優於現有文獻上的建議)、降低了電腦模擬時記憶體的實際需求、而且特別適用於符號機率迅速衰減的分佈。

關鍵詞：聯合訊號源編碼和通道編碼、分離訊號源及通道編碼、不定長度之錯誤更正碼、符號錯誤率。

一、前言

近年來無線通訊與資訊大爆發的時代來臨，對於通訊系統的要求不外乎是更穩定的傳輸、更高的頻譜效益、更高的系統效能等等是大多數討論度非常高的研究方向，在現代數位通信系統中，一開始先使用訊號源執行來源訊號編碼方式可以有效地對來源訊號數據進行取樣，並使用訊號源編碼進行資料壓縮（Data Compression）。接下來，在傳輸過程中為了對抗衰落和雜訊的干擾，則使用通道編碼的方式，因此，建構能趨近通道容量之最佳碼是研究通道編碼設計主要目標。「分離原理」是 Shannon [1]在 1948 年所提出的重要原理，該原理指出來源和通道編碼方案的過程可以分開設計和實現，而整個系統的最佳化是不會被影響，所以理想情況下，一般是不會有任何損失的，可以看成是個別獨立的部分來設計考量，但是，上述系統分開設計，理論上是不會影響系統傳輸的最佳化，但是需要事先假設有二大先決條件，第一就是要假設系統有提供無限大複雜度的先決條件，第二是系統編碼延遲也可以有無限大的等待時間，基於上述二點重要的假設，在現今資源嚴重受限的條件下，這條件假設是非常不切實際也不可能會發生的，所以，該結論的最佳化應該屬於是漸近性的現象。

結合訊號源與通道的聯合編碼（Joint Source-Channel Coding, JSCC）編碼背後的主要思想是，對於固定大小的區塊，執行編碼或非編碼源是存在些許冗餘的，並且這種冗餘可以在解碼器側加以利用。研究表明，對於系統具有嚴格延遲和硬體複雜性限制的情況下，結合訊號源與通道的聯合編碼方案[2]-[5]，將來源編碼和通道編碼方案的設計組合成一個單階策略，可進一步改善資料壓縮率與編碼率的綜合表現，在許多實際應用中，其性能優於「分開設計的來源編碼和通

道編碼（Separate Source-Channel Coding，SSCC）」方案。

在這篇文章中，最佳的「結合訊號源與通道的聯合編碼」定義是能夠保證其錯誤更正能力，並且同時保持冗餘的能力，並盡可能保持較低的碼字長度。不定長度的錯誤更正碼 (variable-length error-correcting codes，VLEC codes) [2]-[6]，具備了二項特性：一、具有將較短長度的碼字分配給更頻繁出現的來源符號，因此降低了碼的平均碼字長度 (average codeword length，ACL)；二、編碼的結構能夠進行錯誤更正，在[4]中顯示，自由距離(free distance，d_{free})是決定 VLEC 碼的錯誤更正能力的關鍵參數。

對於 VLEC 碼的設計，碼字長度的分佈應該與來源字母的出現機率統計互相匹配，這樣才能建構出接近最佳的碼，為了設計最佳的演算法，第一個碼字長度l_1的選擇，是一個關鍵問題，假設l_1選擇範圍給予太大的空間，則由於l_i值屬於單調遞增序列，因此平均碼字長度將會太大而無法實現接近最佳的設計。但是，如果l_1選擇範圍給予過小，則可能無法成功構造出合格的 VLEC 碼簿。在本篇文章中，我們的目標是證明出最短的碼字長度l_1，因此，可以協助演算法構建接近最佳的 VLEC 碼，並專注於各種不同的自由距離約束下之最小的 ACL 值。

二、系統描述

假設有 M 個離散無記憶來源的字母 $A=\{\alpha_1,\alpha_2,...,\alpha_M\}$，其相對應的機率值給定 $p_1,p_2,...,p_M$，$\sum_{i=1}^{M}p_i=1$，VLEC 碼簿為二進制表示之 $C=\{\mathbf{c}_1,\mathbf{c}_2,...,\mathbf{c}_M\}$，$\mathbf{c}_i$將對應到$\alpha_i$，然而碼長長度定義成$l_i$，定義 ACL 的計算公式為$\omega=\sum_{i=1}^{M}p_i\cdot l_i$。

自由距離的定義，free-distance 是不定長度錯誤更正碼中最重要的核心參數，它是影響解碼錯誤率的主要原因，假設接收端知道接收到字元符號的數量是 L 與位元數的數量長度是 N，$X_{L,N}$ 為 L 與 N 所形成的序列集合，其定義為：

$$X_{L,N}\equiv\left\{\underline{\mathbf{x}}=(\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_L):\forall\mathbf{x}_i\in C,\ \sum_{i=1}^{L}\left|l_{\mathbf{x}_i}\right|=N\right\}. \tag{1}$$

Buttigieg[4]，首先將自由距離定義為網格中相同節點上聚會的任意兩個不同序列$\underline{\mathbf{x}}$和$\underline{\mathbf{y}}$之間的最小漢明距離，自由距離的定義如下式：

$$d_{\text{free}}(C)=\min\left\{d_{\text{h}}(\underline{\mathbf{x}},\underline{\mathbf{y}}):\underline{\mathbf{x}},\underline{\mathbf{y}}\in X_N \text{ for some } N \text{ and } \underline{\mathbf{x}}\neq\underline{\mathbf{y}}\right\}. \tag{2}$$

其中$d_{\text{h}}(\underline{\mathbf{x}},\underline{\mathbf{y}})$表示具有相同長度的序列碼$\underline{\mathbf{x}}$和$\underline{\mathbf{y}}$之間的漢明距離。另外[12]在接收器處假設 L 和 N 的訊息是已知且可用的，將其定義修改為：

$$d_{\text{free}}(C)=\min\left\{d_{\text{h}}(\underline{\mathbf{x}},\underline{\mathbf{y}}):\underline{\mathbf{x}},\underline{\mathbf{y}}\in X_{L,N} \text{ for some } L,N \text{ and } \underline{\mathbf{x}}\neq\underline{\mathbf{y}}\right\}. \tag{3}$$

然而，當 N 的長度持續增大時，網格內的路徑數量會呈現指數性增加，因此增加了計算的困難度，所以 Buttigieg 做出了證明了 free-distance 必需要滿足以下的數學式，他讓原本是要龐大的路徑數量組合出的序列$\underline{\mathbf{x}}$才能去計算 free-distance，轉換成只要碼字與碼字間符合數學式（4），即可滿足當下的 free-distance，所以後續大部分的研究都以下列的數學式做為分析 VLEC 的基本規則，d_{free}必需要滿足：

$$d_{\text{free}}(C)\geq\min\left\{d_{\text{eq}}(C),d_{\text{div}}(C)+d_{\text{conv}}(C)\right\}. \tag{4}$$

在[4]中$d_{\text{eq}}(C)$表示具有相同長度碼字之間的漢明距離計算：

$$d_{\text{eq}}(C)=\min\left\{d_{\text{h}}(\mathbf{c}_i,\mathbf{c}_j):(\mathbf{c}_i,\mathbf{c}_j)\in C,\ \mathbf{c}_i\neq\mathbf{c}_j,\text{ and } l_i=l_j\right\}. \tag{5}$$

接著，$d_{\text{div}}(C)$是代表長度不相等的碼字，取碼字前段相等長度(靠左對齊）執行漢明距離計算，稱為最小發散距離(minimum diverging distance)，並且定義如下：

$$d_{\text{div}}(C)=\min\left\{d_{\text{h}}\left(c_i^1c_i^2...c_i^{l_j},c_i^1c_i^2...c_i^{l_j}\right):\left(\mathbf{c}_i,\mathbf{c}_j\right)\in C,\right.$$
$$\left.\text{and } l_i>l_j\right\},\text{ with } \mathbf{c}_i=\left(c_i^1c_i^2...c_i^{l_i}\right). \tag{6}$$

最後，$d_{\text{conv}}(C)$是代表長度不相等的碼字，取碼字後段相等長度(靠右對齊）執行漢明距離計算，稱為最小收斂距離(minimum converging distance)，其定義如下：

$$d_{\text{conv}}(C)=\min\left\{d_{\text{h}}\left(c_i^{l_i-l_j+1}c_i^{l_i-l_j+2}...c_i^{l_i},c_i^1c_i^2...c_i^{l_j}\right):\right.$$
$$\left.\left(\mathbf{c}_i,\mathbf{c}_j\right)\in C,\text{ and } l_i>l_j\right\} \tag{7}$$

在[4]中顯示，自由距離d_{free}是 VLEC 碼最重要的因素，它主導在高 SNR 區域中錯誤更正的能力，此外，[4]也證明了，令人滿意的編碼中 d_{div} 和 d_{conv} 還需要去滿足$\left|d_{\text{div}}(C)-d_{\text{conv}}(C)\right|\le 1$。此外，為了有效地接近最小 ACL 值，我們限制相鄰碼字長度之間的差異，使其小於或等於 1。因此，我們加入距離約束：

$$\begin{aligned}&\min\left\{d_{\text{eq}}(C),d_{\text{div}}(C)+d_{\text{conv}}(C)\right\}\ge d_{\text{free}}^*,\\&\left|d_{\text{div}}(C)-d_{\text{conv}}(C)\right|\le 1,\\&\left|l_i-l_{i+1}\right|\le 1,\ i=1,2,...,M-1.\end{aligned} \tag{8}$$

因此，我們的焦點是依數學式(8)建置接近最佳的 VLEC 碼，並專注於各種不同的自由距離約束下之最小的 ACL 值。

三、加速構建 VLEC 代碼的特性和標準

對於 VLEC 碼的設計，碼字長度的分佈應該與來源字母出現機率的統計互相匹配，這樣才能建構出接近最優的碼，為了設計最佳的演算法，第一個碼字長度l_1的選擇，是一個關鍵問題。考慮到確定l_1的值，我們有以下證明：

規範 1：假設我們現在要去建構前 3 個合格 VLEC 的碼字$\left(\mathbf{c}_1,\mathbf{c}_2,\mathbf{c}_3\right)$，碼字長度的限制$l_1\le l_2\le l_3$，我們首先設定$l_1=d_{\text{free}}^*-x$，$x$表示為正整數。如果要成功的被建構，當$d_{\text{free}}^*$為奇數時，成立條件符合$x\le\dfrac{d_{\text{free}}^*-1}{4}$。

證明：首先，我們先強調接下來的證明都是根據(8)為主要的參考點，我們雖然採用$d_{\text{d}}\left(\mathbf{c}_1,\mathbf{c}_2\right)=d_{\text{c}}\left(\mathbf{c}_1,\mathbf{c}_2\right)+1$為主要的證明參照，但是，當$d_{\text{c}}\left(\mathbf{c}_1,\mathbf{c}_2\right)=d_{\text{d}}\left(\mathbf{c}_1,\mathbf{c}_2\right)+1$也是可以成立的。接下來進行符號定義，符號⊕定義為互斥或閘 XOR(exclusive-or operator)，符號□定義為反互斥或閘 XNOR（exclusive-nor operators)。

我們先設置 VLEC 的第 1 個及第 2 個碼字，分別為$\mathbf{c}_1$及$\mathbf{c}_2$，因為$l_1=d_{\text{free}}^*-x<d_{\text{free}}^*$，這個限制很明確的告知$l_1\ne l_2$，因為當$l_1=l_2$時，必須滿足條件$d_{\text{h}}\left(\mathbf{c}_1,\mathbf{c}_2\right)\ge d_{\text{free}}^*$，基於這個限制，我們只允許考慮$l_1<l_2$，$\mathbf{c}_2$的長度必為$l_2=d_{\text{free}}^*-x+1$。接下來，定義$\mathbf{c}_2$的前置碼與$\mathbf{c}_1$長度$l_1$相等，$\mathbf{c}_2^{\text{p},1}\equiv\left(c_2^1,c_2^2,\cdots,c_2^{l_1}\right)$，因此得到：

$$d_{\text{d}}\left(\mathbf{c}_1,\mathbf{c}_2\right)=w\left(\mathbf{c}_1\oplus\mathbf{c}_2^{\text{p},1}\right). \tag{9}$$

如果$\mathbf{c}_2$是當前有效的候選 VLEC 碼字，它必定滿足$d_{\text{d}}\left(\mathbf{c}_1,\mathbf{c}_2\right)\ge\dfrac{d_{\text{free}}^*+1}{2}$，$w\left(\mathbf{c}_1\square\ \mathbf{c}_2^{\text{p},1}\right)$表示$\mathbf{c}_1$和$\mathbf{c}_2^{\text{p},1}$之間相同位元的數量並滿足$w\left(\mathbf{c}_1\square\ \mathbf{c}_2^{\text{p},1}\right)=l_1-w\left(\mathbf{c}_1\oplus\mathbf{c}_2^{\text{p},1}\right)\le d_{\text{free}}^*-x-\dfrac{d_{\text{free}}^*+1}{2}$。我們考慮$l_i<l_j$並更進一步定義位元索引集合的符號$\square_{i,j}$和$\square_{i,j}^{s}$，它們分別代表$\mathbf{c}_i$和$\mathbf{c}_j^{\text{p},i}$不一樣和一樣的二進制值。

現在要搜索$\mathbf{c}_3$，因為$l_2=d_{\text{free}}^*-x+1\le d_{\text{free}}^*$，有二種可能需要被考慮，說明如下：

方案 1：$l_2=d_{\text{free}}^*$ $(i.e., x=1)$

按照上述類似的定義，我們首先定義$\mathbf{c}_3$的前置編碼，長度l_1及l_2，分別是

$\mathbf{c}_3^{\mathrm{p},1} \equiv \left(c_3^1,\ c_3^2,\ \cdots,c_3^{l_1}\right)$ $\mathbf{c}_3^{\mathrm{p},2} \equiv \left(c_3^1,\ c_3^2,\ \cdots,c_3^{l_2}\right)$，假設 $l_2 = d_{\mathrm{free}}^*$，為了得到最短的平均碼長，我們打算尋找 $\mathbf{c}_3$ 的碼字長度 $l_2 = l_3 = d_{\mathrm{free}}^*$，根據上述公式(8)的限制條件，得到 $\mathbf{c}_3^{\mathrm{p},2} = \mathbf{c}_3$ 和 $d_{\mathrm{h}}\left(\mathbf{c}_2, \mathbf{c}_3\right) = d_{\mathrm{free}}^*$，因此，$\mathbf{c}_2$ 和 $\mathbf{c}_3$ 互為反相，換句話說，$\mathbf{c}_2$ 的所有位元的相反會剛好與 $\mathbf{c}_3$ 的位元相同，因此，考慮到 $\square_{1,2}$ 位元中包含的位置會相等於 $\mathbf{c}_3^{\mathrm{p},1}$，因此，我們把 $\mathbf{c}_1$ 和 $\mathbf{c}_3^{\mathrm{p},1}$ 碼字距離寫成數學式等效於

$$
\begin{aligned}
w\left(\mathbf{c}_1 \oplus \mathbf{c}_3^{\mathrm{p},1}\right) &= w\left(\mathbf{c}_1 \square\ \mathbf{c}_2^{\mathrm{p},1}\right) \le \left(d_{\mathrm{free}}^* - x\right) - \frac{d_{\mathrm{free}}^* + 1}{2} \\
&= \frac{d_{\mathrm{free}}^* - 3}{2} < \frac{d_{\mathrm{free}}^* + 1}{2}
\end{aligned} \tag{10}
$$

，從數學式的結果呈現，$w\left(\mathbf{c}_1 \oplus \mathbf{c}_3^{\mathrm{p},1}\right) < \dfrac{d_{\mathrm{free}}^* + 1}{2}$，當我們去給定 $l_2 = l_3$ 是絕對不可能成功建構 $\mathbf{c}_3$，所以，我們只能假設 $l_2 < l_3$ 的情況下搜索 $\mathbf{c}_3$。

方案 2： $l_2 < d_{\mathrm{free}}^*$ $(i.e.,\ x>1)$

當 $l_2 < d_{\mathrm{free}}^*$，這在 $l_2 = l_3$ 的情況下，無法建構 $\mathbf{c}_3$，因為它無法滿足條件 $w\left(\mathbf{c}_2 \oplus \mathbf{c}_3\right) = d_{\mathrm{free}}^*$，基於上述的原因，我們要成功建構 $\mathbf{c}_3$，只能允許條件符合 $l_2 < l_3$。

根據上述方案 1 及方案 2 的結論得知，在 $l_1 = d_{\mathrm{free}}^* - x < d_{\mathrm{free}}^*$ 的假設條件下，要成功的建構合格的 VLEC 碼字 $\mathbf{c}_3$，條件必滿足為 $l_2 < l_3$，所以 $l_3 = d_{\mathrm{free}}^* - x + 2$，因此，$\mathbf{c}_3$ 要滿足下列的二種情況：

$$
w\left(\mathbf{c}_1 \oplus \mathbf{c}_3^{\mathrm{p},1}\right) \ge \frac{d_{\mathrm{free}}^* + 1}{2}, \tag{11}
$$

$$
w\left(\mathbf{c}_2 \oplus \mathbf{c}_3^{\mathrm{p},2}\right) \ge \frac{d_{\mathrm{free}}^* + 1}{2}. \tag{12}
$$

我們現在要來討論 $\square_{1,2}$ 位元位置給定的情況，假設 $\mathbf{c}_3^{\mathrm{p},1}$ 會有某幾個位元的位置值與 $\mathbf{c}_1$ 不相同，然而，這些不相同的位置，恰好會與 $\mathbf{c}_2^{\mathrm{p},1}$ 相同，換一種方式來說明，$\square_{1,2}$ 所包含的位置，對 $\mathbf{c}_3^{\mathrm{p},1}$ 而言，在相同位置不可能同時不同於 $\mathbf{c}_1$ 及 $\mathbf{c}_2^{\mathrm{p},1}$，相較之下，我們關注 $\square_{1,2}^{s}$ 位元位置的情形，對 $\mathbf{c}_3^{\mathrm{p},1}$ 而言，在相同位置上同時不相同於 $\mathbf{c}_1$ 及 $\mathbf{c}_2^{\mathrm{p},1}$，最後，對 $\mathbf{c}_3^{\mathrm{p},1}$ 而言，在 $\square_{1,2}^{s}$ 即使所有位置都剛好不同於 $\mathbf{c}_1$ 及 $\mathbf{c}_2^{\mathrm{p},1}$ 以滿足數學式(11)，我們仍然需要在相同位置的 $\square_{1,2}$ 中，取「λ」數量的位元位置去不同於 $\mathbf{c}_1$，數學式解析如下：

$$
\begin{aligned}
\lambda &= w\left(\mathbf{c}_1 \oplus \mathbf{c}_3^{\mathrm{p},1}\right) - \left|\square_{1,2}^{s}\right| \ge \frac{d_{\mathrm{free}}^* + 1}{2} - \left(l_1 - \square_{1,2}\right) \\
&\ge \frac{d_{\mathrm{free}}^* + 1}{2} - \left(d_{\mathrm{free}}^* - x\right) + \frac{d_{\mathrm{free}}^* + 1}{2} \\
&= x + 1.
\end{aligned} \tag{13}
$$

基於上述的結果，在 $\square_{1,2}$ 的區域位置中，$\mathbf{c}_3$ 會有 λ 個位元位置值與 $\mathbf{c}_1$ 不相同，然而，它就會有 λ 個位元位置值與 $\mathbf{c}_1$ 相同，接下來，我們要再額外定義「δ」數量的位元位置，它表示在 $\square_{1,2}$ 中，$\mathbf{c}_2$ 與 $\mathbf{c}_3^{\mathrm{p},2}$ 不相同位元位置的數量。此外，因為 $\mathbf{c}_2$ 的長度會比 $\mathbf{c}_1$ 長 1 個位元，所以這個位元是一個額外的差異位元在 $\mathbf{c}_2$ 和 $\mathbf{c}_3^{\mathrm{p},2}$ 的區域範圍內，最後，$w\left(\mathbf{c}_2 \oplus \mathbf{c}_3^{\mathrm{p},2}\right)$ 可以被寫成

$$
\begin{aligned}
w\left(\mathbf{c}_2 \oplus \mathbf{c}_3^{\mathrm{p},2}\right) &= \delta + \left|\square_{1,2}^{s}\right| + 1 \\
&= \left|\square_{1,2}\right| - \lambda + \left|\square_{1,2}^{s}\right| + 1 = l_1 - \lambda + 1 \\
&\le \left(d_{\mathrm{free}}^* - x\right) - \left(x + 1\right) + 1 = d_{\mathrm{free}}^* - 2x.
\end{aligned} \tag{14}
$$

因此，為了滿足數學式（12），我們可以寫成 $d_{\mathrm{free}}^* - 2x \ge \dfrac{d_{\mathrm{free}}^* + 1}{2}$，得到 $x \le \dfrac{d_{\mathrm{free}}^* - 1}{4}$，所以從上述的結果得知，只要 $x > \dfrac{d_{\mathrm{free}}^* - 1}{4}$，保證無法去取得合格的碼字 $\mathbf{c}_3$，VLEC 碼簿就絕對不可能被建置成功。

結論，當 d_{free}^* 為奇數時，要取得 3 個（含）以上的合格 VLEC 碼字，$\mathbf{c}_1$ 的最短長度

$l_1 = d^*_{\text{free}} - x = d^*_{\text{free}} - \frac{d^*_{\text{free}} - 1}{4} = \frac{3d^*_{\text{free}} + 1}{4}$ 。

我們現在提供一個例子如表 1 來做說明，當 d^*_{free} 設定它等於 9，在 Codebook A1 和 A2，我們分別選擇 $l_1 = d^*_{\text{free}} - 2 = 7\left(x = 2\right)$ 和 $l_1 = d^*_{\text{free}} - 3 = 6\left(x = 3\right)$。對 Codebook A1 而言，$x = 2 \le \frac{d^*_{\text{free}} - 1}{4} = 2$，所以合格的候選碼字 $\mathbf{c}_3$ 是可以被發掘的，然而，對 Codebook A2 而言，$x = 3 > \frac{d^*_{\text{free}} - 1}{4} = 2$，是絕對不可能有合格的候選碼字 $\mathbf{c}_3$ 是可以被發掘的，因此 VLEC 碼簿是無法被成功建置的。

表 1 規範 1 的例子說明

	Codebook A1 $d^*_{\text{free}} = 9, l_1 = 7$	**Codebook A2** $d^*_{\text{free}} = 9, l_1 = 6$
$\mathbf{c}_1$	0000111	001111
$\mathbf{c}_2$	00110001	0100001
$\mathbf{c}_3$	110000000	10000000(無效的)
	⋮	失敗

準則 1：為了找尋最小化 ACL 值，我們打算限制第一個碼字的長度從最小的 l_1 值開始構建，當 d^*_{free} 為奇數時，我們不需要浪費時間去搜尋 $l_1 < \frac{3d^*_{\text{free}} + 1}{4}$，因此，可以縮小搜索空間、增加搜索速度。

規範 2：當 d^*_{free} 為奇數時，假設我們現在要去建構 3 個連續的合格 VLEC 的碼字 $\left(\mathbf{c}_i, \mathbf{c}_{i+1}, \mathbf{c}_{i+2}\right)$，碼字長度 $l_i \le l_{i+1} \le l_{i+2}$，假設碼字長度 $l_i = l_{i+1} = d^*_{\text{free}}$，合格的候選碼字 $\mathbf{c}_{i+2}$ 是不可能存在的。

證明：我們優先考慮 VLEC 碼字 $\mathbf{c}_i$ 和 $\mathbf{c}_{i+1}$，假設我們令碼字長度 $l_i = l_{i+1} = d^*_{\text{free}}$，對 $\mathbf{c}_{i+1}$ 而言，所有的二進度位元只允許與 $\mathbf{c}_i$ 相反，也就是呈現反相狀態，因為 $l_i = l_{i+1} = d^*_{\text{free}}$，所以得到 $w\left(\mathbf{c}_i \oplus \mathbf{c}_{i+1}\right) = d^*_{\text{free}}$。接下來，當要合格的候選碼字 $\mathbf{c}_{i+2}$，假設我們只考慮到 $l_{i+2} = l_i = l_{i+1}$ 的條件 $w\left(\mathbf{c}_i \oplus \mathbf{c}_{i+1}\right) = w\left(\mathbf{c}_i \oplus \mathbf{c}_{i+2}\right) = w\left(\mathbf{c}_{i+1} \oplus \mathbf{c}_{i+2}\right) = d^*_{\text{free}}$ 就一定會被滿足，但是，很明確地，這保證不可能成功建置 $\mathbf{c}_{i+2}$ 的，因為 $\mathbf{c}_{i+2}$ 的每個二進制位元不可能同時相異於 $\mathbf{c}_i$ 和 $\mathbf{c}_{i+1}$。因此，當碼字長度 $l_i = l_{i+1} = d^*_{\text{free}}$ 時，$\mathbf{c}_{i+2}$ 若要成功被建構，只允許條件 $l_{i+2} > l_i = l_{i+1} = d^*_{\text{free}}$。

假設條件 $l_{i+2} > l_i = l_{i+1} = d^*_{\text{free}}$，因為 $l_i = l_{i+1}$，條件 $\mathbf{c}^{\text{p},i}_{i+2} = \mathbf{c}^{\text{p},i+1}_{i+2}$ 是滿足的，若 $\mathbf{c}_{i+2}$ 要成為合格的候選碼字，它需要同時滿足 $w\left(\mathbf{c}_i \oplus \mathbf{c}^{\text{p},i}_{i+2}\right) \ge \frac{d^*_{\text{free}} + 1}{2}$ 和 $w\left(\mathbf{c}_{i+1} \oplus \mathbf{c}^{\text{p},i+1}_{i+2}\right) \ge \frac{d^*_{\text{free}} + 1}{2}$，我們現在假設第 1 個條件是成立的，因此，$\mathbf{c}_i$ 和 $\mathbf{c}_{i+2}$ 相異的二進制位元數總和為 $w\left(\mathbf{c}_i \oplus \mathbf{c}^{\text{p},i}_{i+2}\right) \ge \frac{d^*_{\text{free}} + 1}{2}$，接下來，我們去計算 $w\left(\mathbf{c}_{i+1} \oplus \mathbf{c}^{\text{p},i+1}_{i+2}\right) \le d^*_{\text{free}} - \frac{d^*_{\text{free}} + 1}{2} = \frac{d^*_{\text{free}} - 1}{2}$，所以公式(15)不可能被滿足，我們是無法成功建構 $\mathbf{c}_{i+2}$。

準則 2：所以對於碼字長度的確定，我們有以下準則：當 d^*_{free} 為奇數且 $l_i \le d^*_{\text{free}}$，準備建構 $\mathbf{c}_{i+1}$，對於碼字長度 l_{i+1} 的選擇，必需要符合 $l_{i+1} > l_i$。

規範 3：假設當 d^*_{free} 為奇數且考慮 $l_1 = l_2 = d^*_{\text{free}} + 1$，現在要去選擇碼字 $\mathbf{c}_2$，當我們選取 $d_{\text{h}}\left(\mathbf{c}_1, \mathbf{c}_2\right) = d^*_{\text{free}}$，這意指在碼字 $\mathbf{c}_1$ 和 $\mathbf{c}_2$ 的二進制位元位置上，有著某一個位置會有相同「0」或「1」，然而，非常湊巧的，當持續往下建構 VLEC 碼簿時，這一個特定位置會永遠出現「1」或「0」。

證明：接下來這個案例，我們使用表 2 的例子來做說明，在 Codebook B1 中，設定

$d_{\text{free}}^{*}=5$ 及第 1 個碼字長度和第 2 個碼字長度 $l_1=l_2=d_{\text{free}}^{*}+1=6$，不失一般性，我們設定 $\mathbf{c}_1$ 和 $\mathbf{c}_2$ 最左邊的位元位置，同時出現「0」，接下來要決定 $\mathbf{c}_3$，假設我們同樣設定 $l_3=d_{\text{free}}^{*}+1=6$，這是不可能去完成建置的，因為 $\mathbf{c}_1$ 和 $\mathbf{c}_2$ 的最小漢明距離是 $d_{\text{free}}^{*}=5$，不可能在漢明距離為 d_{free}^{*} 且碼字長度為 $d_{\text{free}}^{*}+1$ 的情況下，出現 3 組同樣長度的碼字，所以，必需假定 $l_3=d_{\text{free}}^{*}+2$ 且滿足 $d_{\text{d}}\left(\mathbf{c}_3,\mathbf{c}_1\right)=w\left(\mathbf{c}_3\oplus\mathbf{c}_2\right)\geq\frac{d_{\text{free}}^{*}+1}{2}$。因為設定 $d_{\text{free}}^{*}=5$，所需距離的最小值正好是 3，結果表明，要滿足此約束條件，$\mathbf{c}_3$ 的最左邊位元位置必須為「1」，類似地原理，由於與 $\mathbf{c}_1$ 和 $\mathbf{c}_2$ 相比較，所有剩餘的碼字也應滿足相同的約束，因此剩餘碼字的最左邊的位元位置將始終為「1」。

表 2 規範 3 的例子說明

	Codebook B1 $\left(d_{\text{div}}^{*},d_{\text{conv}}^{*}\right)=\left(3,2\right)$	**Codebook B2** $\left(d_{\text{div}}^{*},d_{\text{conv}}^{*}\right)=\left(2,3\right)$
$\mathbf{c}_1$	**0**00111	**0**00100
$\mathbf{c}_2$	**0**11000	**0**11011
$\mathbf{c}_3$	**1**00000**0**	**11**00010
$\mathbf{c}_4$	**1**11111**0**	**11**11101
$\mathbf{c}_5$	**1**01001**1**0	0**01**10111
$\mathbf{c}_6$	**1**10110**1**1	1**01**01000
$\mathbf{c}_7$	**1**01010**1**1**0**	**1**0**01**01110
$\mathbf{c}_8$	**1**10101**1**0**0**	**1**1**01**10001
$\mathbf{c}_9$	**1**01100**1**1**1**0	0**0**0**01**10000
$\mathbf{c}_{10}$	**1**10011**1**0**1**1	0**01**0**1**01111
$\mathbf{c}_{11}$	**1**10001**1**1**1**00	01**0**1**01**11110
$\mathbf{c}_{12}$	101110101111	001000100001

由規範 3 得知，當我們要為表 2 給出的 Codebook B1 確定 $\mathbf{c}_4$ 時，$\mathbf{c}_4$ 的最左邊位元位置已經顯示為「1」。結果，如果我們讓碼字長度 l_4 為 $d_{\text{free}}^{*}+2$，那麼在 $\mathbf{c}_4$ 的設計中“可用”位元位置的數量實際上只有 $d_{\text{free}}^{*}+1$, 因此，如果我們仍然讓 $\mathbf{c}_3$ 和 $\mathbf{c}_4$ 在 $d_{\text{free}}^{*}+1$ 個可用位元位置中，有著 d_{free}^{*} 位元位置不相同，則會出現與規範 3 中提到的類似情況。在 Codebook B1 中觀察到，從 $\mathbf{c}_5$ 開始，對於剩餘的碼字位置，從左側起第 7 位元位置的位值將始終為「1」。因此，可以得出結論，如果出現規範 3 中考慮的情況，則在構造 $\mathbf{c}_{i+1}$ 時，“可用”的位元位置的數量將始終為 $d_{\text{free}}^{*}+1$，在 i 為奇數時。在這種情況下，如果我們總是讓 $l_{i+1}=l_i$ 和 $d_{\text{h}}\left(\mathbf{c}_{i+1},\mathbf{c}_i\right)=d_{\text{free}}^{*}$，當 $l_{i+2}=l_{i+1}=l_i$ 則不可能建構出 $\mathbf{c}_{i+2}$，因此，我們必須將碼字長度 l_{i+2} 增加 1 個位元，因此，每兩個碼字，所建構的碼字的長度將會持續增加 1 位元。從表 2 中的 Codebook B1 和表 3 中的 Codebook C1 的比較可以看出，B1 中長度相同的碼字是不會超過兩個以上，因此，將會導致產生較大的 ACL 值。最後，值得一提的是，即使我們改變本文中考慮的假設並讓 $d_{\text{c}}\left(\mathbf{c}_i,\mathbf{c}_j\right)=d_{\text{d}}\left(\mathbf{c}_i,\mathbf{c}_j\right)+1$，如 Codebook B3 所示，連續個「1」仍然出現，只是持續的向右移動呈現。

為了避免在規範 3 中觀察到的情況，一個直觀的建構策略是，如果我們現在已有 $\mathbf{c}_i$ 且 $l_i>l_{i-1}$，將決定 $\mathbf{c}_{i+1}$ 並確定滿足 $l_{i+1}=l_i$，因此將會決定去選擇 $d_{\text{h}}\left(\mathbf{c}_{i+1},\mathbf{c}_i\right)=l_i$。如果我們遵循這個策略，我們就建立了以下規範。

規範 4：假設已經構建了 VLEC 碼字 $\mathbf{c}_i$ 及 $l_i>l_{i-1}$，當確定滿足長度 $l_i=l_{i+1}$ 的 $\mathbf{c}_{i+1}$ 值時，假設我們永遠選擇 $d_{\text{h}}\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=l_i$，它不可能找到具有相同長度的後續碼字 $\mathbf{c}_{i+2}$，除非長度為 $l_i=l_{i+1}=l_{i+2}\geq 2d_{\text{free}}^{*}$。

證明：當參數 τ 等於 x 時，規範 4 是規範 5 的特例，詳情請參考規範 5 的證明。

為了避免在規範 3 中觀察到的情況，當長

度為 $l_{i-1} < l_i = l_{i+1}$ 時，我們盡可能希望讓 $\mathbf{c}_i$ 和 $\mathbf{c}_{i+1}$ 在所有的位元位置上都不相同，然而，規範 4 揭示了我們無法確定與 $\mathbf{c}_i$ 和 $\mathbf{c}_{i+1}$ 具有相同長度的第 3 個碼字 $\mathbf{c}_{i+2}$，因此，我們需將碼字的長度增加 1 位元，因此，類似於規範 3 中建議的情況，構建的碼字長度仍然繼續於每兩個碼字增加 1 位元，直到達到 $2d^*_{\text{free}}$ 的值，因此，ACL 值將會顯著的增加。為了解決這個問題，我們提出以下建議：

表 3 規範 4 的例子說明

	Codebook C1 $\left(d^*_{\text{div}}, d^*_{\text{conv}}\right) = (3,2)$	**Codebook C2** $\left(d^*_{\text{div}}, d^*_{\text{conv}}\right) = (4,3)$
$\mathbf{c}_1$	001100	00000000
$\mathbf{c}_2$	110011	11111111
$\mathbf{c}_3$	0100000	001110100
$\mathbf{c}_4$	1011111	110001011
$\mathbf{c}_5$	00000111	0000111110
$\mathbf{c}_6$	11111000	1111000001
$\mathbf{c}_7$	000111001	01010110010
$\mathbf{c}_8$	111000110	10101001101
$\mathbf{c}_9$	**0101101000**	010100111001
$\mathbf{c}_{10}$	**1010010111**	101011000110
$\mathbf{c}_{11}$	**0110101111**	0101100100110
$\mathbf{c}_{12}$	**1001010000**	1010011011001

規範 5： 假設當 d^*_{free} 為奇數時，我們先定義 $d^*_{\text{div}} = \dfrac{d^*_{\text{free}}+1}{2}$ 及 $d^*_{\text{conv}} = d^*_{\text{free}} - d^*_{\text{div}}$ 條件成立，接下來，我們想要去建構 3 個（含）以上的相同長度之合格 VLEC 碼字，它需要符合並滿足以下條件 $l_i - \max\left(d^*_{\text{div}}, d^*_{\text{conv}}\right) \geq d^*_{\text{free}}$，才有機會成功完成建構。

證明： 我們現在假設 $\mathbf{c}_i$ 及 $\mathbf{c}_{i+1}$ 是可以被建構的，長度 $l_{i-1} < l_i$ 及 $l_i = l_{i+1} = d^*_{\text{free}} + x$，$x$ 被表示為正整數。$\Box_{i,j}$ 定義成位元的索引值，它表示 $\mathbf{c}_i$ 和 $\mathbf{c}_j$ 位元不相同的位置值，$\Box^{s}_{i,j}$ 被表示 $\mathbf{c}_i$ 和 $\mathbf{c}_j$ 位元相同的位置值，碼字長度 $l_i = l_j$。我們將引入一個參數 τ，它保證 $\mathbf{c}_i$ 和 $\mathbf{c}_{i+1}$ 之間的設計距離值，因此，$\left|\Box_{i,i+1}\right| = w\left(\mathbf{c}_i \oplus \mathbf{c}_{i+1}\right) \geq d^*_{\text{free}} + \tau$ 及 $\left|\Box^{s}_{i,i+1}\right| \leq x - \tau$，接下來，在 $\Box_{i,i+1}$ 索引位置內，我們設定 $\mathbf{c}'_i$ 和 $\mathbf{c}'_{i+1}$ 分別為 $\mathbf{c}_i$ 及 $\mathbf{c}_{i+1}$ 內的一部分，因此

$$d_{\text{h}}\left(\mathbf{c}_i, \mathbf{c}_{i+1}\right) = w\left(\mathbf{c}_i \oplus \mathbf{c}_{i+1}\right) = w\left(\mathbf{c}'_i \oplus \mathbf{c}'_{i+1}\right) \geq d^*_{\text{free}} + \tau. \tag{16}$$

同理，假設在 $\Box^{s}_{i,i+1}$ 索引位置內，我們設定 $\mathbf{c}''_i$ 和 $\mathbf{c}''_{i+1}$ 分別為 $\mathbf{c}_i$ 和 $\mathbf{c}_{i+1}$ 內的位元相同位置部分，我們可以寫成

$$w\left(\mathbf{c}_i \Box\, \mathbf{c}_{i+1}\right) = w\left(\mathbf{c}''_i \Box\, \mathbf{c}''_{i+1}\right) \leq x - \tau. \tag{17}$$

假設 $\mathbf{c}_{i+2}$ 是一個可以被成功建構的合格 VLEC 碼字，$w\left(\mathbf{c}_i \oplus \mathbf{c}_{i+2}\right) \geq d^*_{\text{free}}$ 和 $w\left(\mathbf{c}_{i+1} \oplus \mathbf{c}_{i+2}\right) \geq d^*_{\text{free}}$ 是保證一定可以被滿足的，我們接下來，設定 $\mathbf{c}'_{i+2}$ 和 $\mathbf{c}''_{i+2}$ 是屬於 $\mathbf{c}_{i+2}$ 位元位置的一部份，$\mathbf{c}'_{i+2}$ 隸屬於 $\Box_{i,i+1}$ 及 $\mathbf{c}''_{i+2}$ 隸屬於 $\Box^{s}_{i,i+1}$，我們接下來觀察 $\mathbf{c}_{i+2}$ 位元位置不同於 $\mathbf{c}_i$ 在 $\Box^{s}_{i,i+1}$ 的範圍內，因此，如果滿足限制約束，我們有

$$\begin{aligned} d_{\text{h}}\left(\mathbf{c}_i, \mathbf{c}_{i+2}\right) &= w\left(\mathbf{c}_i \oplus \mathbf{c}_{i+2}\right) \\ &= w\left(\mathbf{c}''_i \Box\, \mathbf{c}''_{i+1}\right) + w\left(\mathbf{c}'_i \oplus \mathbf{c}'_{i+2}\right) \geq d^*_{\text{free}}, \end{aligned} \tag{18}$$

因此，

$$w\left(\mathbf{c}'_i \oplus \mathbf{c}'_{i+2}\right) \geq d^*_{\text{free}} - w\left(\mathbf{c}''_i \Box\, \mathbf{c}''_{i+1}\right). \tag{19}$$

實際上，對 $\mathbf{c}_{i+2}$ 而言（18）中的假設代表著構建成功的最低要求，在 $\mathbf{c}_i$ 和 $\mathbf{c}_{i+2}$ 之間會有不相同的位元位置是包含在 $\Box_{i,i+1}$ 中的部份，它至少滿足 $d^*_{\text{free}} - w\left(\mathbf{c}''_i \Box\, \mathbf{c}''_{i+1}\right)$。

如果第二個限制條件 $w\left(\mathbf{c}_{i+1} \oplus \mathbf{c}_{i+2}\right) \geq d^*_{\text{free}}$ 也要被滿足，同樣的，我們也可以寫成

$$d_{\mathrm{h}}\left(\mathbf{c}_{i+1},\mathbf{c}_{i+2}\right)=w\left(\mathbf{c}_{i+1}\oplus\mathbf{c}_{i+2}\right)$$
$$=w\left(\mathbf{c}_i^{''}\square\ \mathbf{c}_{i+1}^{''}\right)+w\left(\mathbf{c}_{i+1}^{'}\oplus\mathbf{c}_{i+2}^{'}\right)\geq d_{\mathrm{free}}^{*}, \quad (20)$$

然而，$w\left(\mathbf{c}_{i+1}^{'}\oplus\mathbf{c}_{i+2}^{'}\right)$，它代表在$\square_{i,i+1}$索引位置中$\mathbf{c}_{i+1}$和$\mathbf{c}_{i+2}$不相同的位元數量，剛好又會相同於$w\left(\mathbf{c}_i^{'}\square\ \mathbf{c}_{i+2}^{'}\right)$，$w\left(\mathbf{c}_i^{'}\square\ \mathbf{c}_{i+2}^{'}\right)$代表在$\square_{i,i+1}$索引位置中$\mathbf{c}_i$和$\mathbf{c}_{i+2}$相同的位元數量，我們可以寫成

$$w\left(\mathbf{c}_{i+1}^{'}\oplus\mathbf{c}_{i+2}^{'}\right)=w\left(\mathbf{c}_i^{'}\square\ \mathbf{c}_{i+2}^{'}\right). \quad (21)$$

同樣，我們也可以

$$w\left(\mathbf{c}_i^{'}\oplus\mathbf{c}_{i+2}^{'}\right)=w\left(\mathbf{c}_{i+1}^{'}\square\ \mathbf{c}_{i+2}^{'}\right). \quad (22)$$

基於上述這樣的結果，我們則可以改寫

$$\begin{aligned}d_{\mathrm{h}}\left(\mathbf{c}_{i+1},\mathbf{c}_{i+2}\right)&=w\left(\mathbf{c}_i^{''}\square\ \mathbf{c}_{i+1}^{''}\right)+w\left(\mathbf{c}_{i+1}^{'}\oplus\mathbf{c}_{i+2}^{'}\right)\\&=w\left(\mathbf{c}_i^{''}\square\ \mathbf{c}_{i+1}^{''}\right)+\left|\square_{\mathrm{i,i+1}}\right|-w\left(\mathbf{c}_{i+1}^{'}\square\ \mathbf{c}_{i+2}^{'}\right)\end{aligned} \quad (23)$$

$w\left(\mathbf{c}_i^{''}\square\ \mathbf{c}_{i+1}^{''}\right)+\left|\square_{\mathrm{i,i+1}}\right|$ 會等效於碼長 $l_i=d_{\mathrm{free}}^{*}+x$，考慮（22），我們得到

$$d_{\mathrm{h}}\left(\mathbf{c}_{i+1},\mathbf{c}_{i+2}\right)=d_{\mathrm{free}}^{*}+x-w\left(\mathbf{c}_i^{'}\oplus\mathbf{c}_{i+2}^{'}\right) \quad (24)$$

利用（25）和（27）中的不等式，我們得到

$$\begin{aligned}d_{\mathrm{h}}\left(\mathbf{c}_{i+1},\mathbf{c}_{i+2}\right)&\leq d_{\mathrm{free}}^{*}+x-\left[d_{\mathrm{free}}^{*}-w\left(\mathbf{c}_i^{''}\square\ \mathbf{c}_{i+1}^{''}\right)\right]\\&\leq d_{\mathrm{free}}^{*}+x-\left[d_{\mathrm{free}}^{*}-(x-\tau)\right]=2x-\tau.\end{aligned} \quad (25)$$

整合（19）和（24），因為 $d_{\mathrm{h}}\left(\mathbf{c}_{i+1},\mathbf{c}_{i+2}\right)\geq d_{\mathrm{free}}^{*}$ 是成立的，我們可以寫成，

$$x\geq\frac{d_{\mathrm{free}}^{*}+\tau}{2}. \quad (26)$$

如果我們在$\mathbf{c}_i$及$\mathbf{c}_{i+1}$不增加任何特別的距離值，也就是假設τ等於零，就會得到$d_{\mathrm{h}}\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)\geq d_{\mathrm{free}}^{*}$。再從（25）中得到$x\geq\frac{d_{\mathrm{free}}^{*}}{2}$，因為$x$為正整數，如果$d_{\mathrm{free}}^{*}$設定為奇數時，$x\geq\left\lceil\frac{d_{\mathrm{free}}^{*}}{2}\right\rceil=\frac{d_{\mathrm{free}}^{*}+1}{2}$，最後結果，長度$l_i=d_{\mathrm{free}}^{*}+\max\left(d_{\mathrm{div}}^{*},d_{\mathrm{conv}}^{*}\right)$可以被成立。

從規範 5 印證出，當長度$l_i-\max\left(d_{\mathrm{div}}^{*},d_{\mathrm{conv}}^{*}\right)<d_{\mathrm{free}}^{*}$，無論$\mathbf{c}_{i+1}$如何選擇搭配與$\mathbf{c}_i$長度相同的任何候選碼字，保證都不可能建構到第 3 個相同碼長的合格 VLEC 碼字，因此在這階段，我們選擇$\mathbf{c}_{i+1}$的碼字，就會直接選擇碼字間的漢明距離滿足$w\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=l_i$的這種情況，放棄$w\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)<l_i$，因為選擇$w\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=l_i$是可以提高 VLEC 碼簿的錯誤性能（碼字與碼字間的距離是最大的）。當條件滿足選擇$l_i-\max\left(d_{\mathrm{div}}^{*},d_{\mathrm{conv}}^{*}\right)\geq d_{\mathrm{free}}^{*}$時，我們就可以放寬選碼條件，捨棄選擇$w\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=l_i$的條件，改以$w\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=d_{\mathrm{free}}^{*}+1$為主要首選，當無首選條件時，再以$w\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=d_{\mathrm{free}}^{*}$為次佳之選擇，上述的限制條件除了可以降低平均碼字長度的快速增加，另一方面，也可以增加 VLEC 碼簿的錯誤更正能力。

基於對規範 2-5 的綜合考量，我們總結並提出如下判斷標準：

標準 1：對於當d_{free}^{*}為奇數時，碼字長度$l_i\leq d_{\mathrm{free}}^{*}$的情況，根據規範 2，不能建構滿足碼字長度$l_i=l_{i+1}$的$\mathbf{c}_{i+1}$。

標準 2：對於當d_{free}^{*}為奇數時，且碼字長度$d_{\mathrm{free}}^{*}<l_i=l_{i+1}<d_{\mathrm{free}}^{*}+\max\left(d_{\mathrm{div}}^{*},d_{\mathrm{conv}}^{*}\right)$，我們建議為$\mathbf{c}_{i+1}$選擇一個滿足$d_{\mathrm{h}}\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=l_i$的值。

標準 3：如果碼字長度為$l_i=l_{i+1}\geq d_{\mathrm{free}}^{*}+\max\left(d_{\mathrm{div}}^{*},d_{\mathrm{conv}}^{*}\right)$，則不需要滿足約束條件$d_{\mathrm{h}}\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=l_i$，相反，我們建議為$\mathbf{c}_{i+1}$優先選擇滿足$d_{\mathrm{h}}\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)\geq d_{\mathrm{free}}^{*}+1$，最後再

選擇滿足 $d_{\mathrm{h}}\left(\mathbf{c}_i,\mathbf{c}_{i+1}\right)=d^{*}_{\mathrm{free}}$ 。

四、模擬結果

在本節中，首先，我們考慮了26個符號的英文字母大小的研究結果，文獻[12]演算法實現了目前文獻探討中最低之搜索時間，因此，在表4中，我們和文獻[12]演算法比較搜索時間和電腦使用的記憶體大小，因為我們已經知道 $\mathbf{c}_1$ 編碼的最小長度，可以避免演算法的空轉，不會浪費建置時間，所以我們的建置時間是減少許多的。隨著字母表個數的增加，[12]需要較大的電腦記憶體空間來建構具有可比較的VLEC碼ACL值。我們所提出的演算法僅需要較低的電腦記憶體來儲存使用，演算法實現了隨機搜索方式，使得可以在合理的搜索時間內，搜索到大尺寸字母表的VLEC碼。

表5中列出了概率分佈的VLEC碼本組合，此外，對於英文字母大小的26個符號，使用各種不同構造演算法獲得的VLEC碼本ACL值比較如表6所示，對於所有考慮的不同自由距離值，與文獻中的大多數VLEC碼本相比，本文構建的VLEC碼本在ACL值上實現了顯著改進。作為 SER 性能的演示，我們與之前文獻中的VLEC碼簿相比，圖1表示所提出的碼本實現了最小值和最佳 SER 性能。

表 4 $d^{*}_{\mathrm{free}}=7$ 的建構效率比較

字母個數	文獻[12]			我們的方法		
	記憶體GB	使用時間	ACL	記憶體GB	使用時間	ACL
26	8	33m	10.3615	8	5m	<10.36
32	8	40m	10.0920	8	1m	<10
64	32	5d9h	11.5309	8	3m	<11.8

五、結論

本研究探索了 VLEC 碼的接近最佳之設計，這些 VLEC 碼簿可以壓縮來源數據，同時具有可靠的傳輸，模擬結果表明，使用我們提出的搜索條件可以有效地縮小搜索空間並顯著減少搜索時間。隨機搜索過程算法表明，可以減少具有最小漢明距離值的序列對之數量，因此，我們可以構建接近最佳的VLEC 碼簿，以針對各種自由距離值和英文字母表去降低 ACL 和提高 SER 的性能。

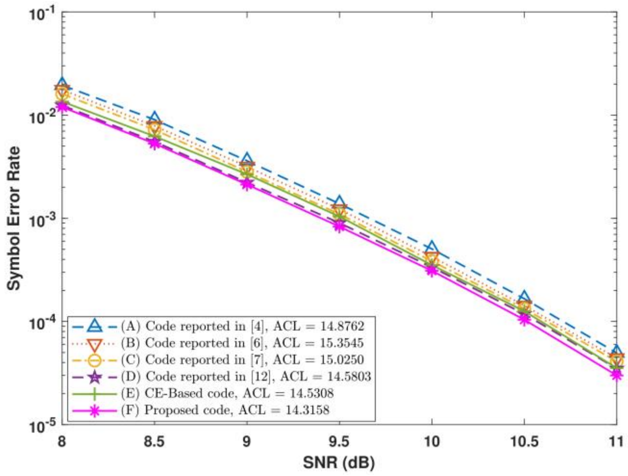

圖 1 不同 VLEC 碼簿對構造 $d^{*}_{\mathrm{free}}=11$ 的 26 個英文字母符號的誤碼性能

表 5 考慮不同自由距離值約束的 26 個英文字母符號的 VLEC 碼簿

Symbol	Probability	$d^*_{\text{free}}=3$ ACL=6.1628	$d^*_{\text{free}}=5$ ACL=8.0535	$d^*_{\text{free}}=7$ ACL=10.2210	$d^*_{\text{free}}=9$ ACL=12.3706	$d^*_{\text{free}}=11$ ACL=14.3159
E	0.148786	0101	001001	00001011	010100011	0000001101
T	0.093541	00000	110110	11110100	0100011001	00011100001
A	0.088337	11110	0001110	101111111	00001111101	011011011001
O	0.072458	001101	1110001	010000000	11111000010	0111101011110
R	0.068722	100110	00111011	0010010111	001000000000	10101000110011
N	0.064985	110001	11000100	1101101000	111111111111	11010111101100
H	0.058313	011000	010000010	10001100010	1001111101001	101100011011110
I	0.056445	0011101	011111100	01111001100	1110001010110	110011101100000
S	0.055378	1001010	100011111	10100010101	10011100001111	1010011010100100
D	0.043768	0110110	101010000	01010111011	01101011000100	1111100101000010
L	0.041233	1010000	111101011	011011101010	10100101110010	10110110110110000
U	0.027622	1000111	000100101	110100111100	111001010011111	11000000101010111
P	0.025754	1100100	0000101000	001110000101	001010101101110	101001001111001111
F	0.024553	10101100	1000000111	100001010011	100111001000001	110100001000010000
M	0.023619	11101011	0111000010	0001011010001	101100110110000	111111110100110111
C	0.020817	00101001	0101011101	0110111001010	0111111010001100	1101101110001110100
W	0.018682	10100111	1011111011	0011000100110	1000101000010111	1110000000110000110
G	0.015212	101010100	1110110101	1110001101101	0011110101100111	1011111111010101010
Y	0.015212	101111010	01001111000	1100110110011	1000100011101000	1100010011111000101
B	0.012677	001011111	01101000110	1001100011100	00110111000111100	1111000000101111000
V	0.011609	111011101	10110000111	11101001011101	11101101100001111	1101010011011110110110
K	0.008674	0010111000	10010111001	11000111101011	00111010111000101	1011101110000000111010
X	0.001468	1011111111	010011101101	00011100101101	10001001011110110	11100100011010100000
J	0.000801	10111110010	100101100100	00100110000110	101010010011010100	11010101110011100110011
Q	0.000801	101111100010	101100010010	00110010011001	110010101000101100	111001100011111111100111
Z	0.000534	101111101001	011010011011	10010001010110	101101100001001010	11110010000000001011010

表 6 考慮不同自由距離約束的 26 個英文字母符號 VLEC 碼的平均碼字長度列表

Algorithm	[4]	[6]	[7]	[12]	交叉熵演算法	建議的演算法
d^*_{free}	ACL	ACL	ACL	ACL	ACL	ACL
3	6.2726	6.3000	6.2666	6.1894	6.1885	6.1628
5	8.3780	8.4010	8.3780	8.3339	8.3077	8.0535
7	10.5596	10.5999	10.4889	10.3025	10.2770	10.2210
9	12.7373	12.8066	12.7373	12.5323	12.5185	12.3706
11	14.8762	15.3545	15.0250	14.5803	14.5308	14.3159

參考文獻

[1] C. E. Shannon, “A mathematical theory of communication,” Bell System Technical J., vol. 27, pt. I, pp. 379-423; pt. II, pp. 623-656, 1948.

[2] F. Alajaji, N. Phamdo, and T. Fuja, “Channel codes that exploit the residual redundancy in CELP-encoded speech,” IEEE Trans. Speech Audio Process., vol. 4, pp. 325-336, Sep. 1996.

[3] M. A. Bernard and B. D. Sharma, “A lower bound on average codeword length of variable length error-correcting codes,” IEEE Transactions on Information Theory, vol. 36, no. 6, pp. 1474-1475, Nov. 1990.

[4] V. Buttigieg, “Variable-length error-correcting codes,” Ph.D. thesis, Univ. of Manchester, England, 1995.

[5] V. Buttigieg and P. G. Farrell, “Variable-length error-correcting codes,” IEEE Proceedings - Communications, vol. 147, no. 4, pp. 211-215, Aug. 2000.

[6] C. Lamy and F. X. Bergot, “Lower bounds on the existence of binary error-correcting variable-length codes,” in Proc. 2003 IEEE Inform. Theory Workshop, pp. 300-303.

[7] J. Wang, L.-L. Yang, and L. Hanzo, “Iterative construction of reversible variable-length codes and variable-length error-correcting codes,” IEEE Commun. Lett., vol. 8, no. 11, pp. 671-673, Nov. 2004.

[8] Y.-M Chen, F.-T. Wu, C.-P. Li and P. K. Varshney, “An efficient construction strategy for near-optimal variable-length error-correcting codes,” IEEE Commun. Lett., vol. 23, no. 3, pp. 398–401, Mar. 2019.

[9] Y.-M Chen, F.-T. Wu, C.-P. Li and P. K. Varshney, “On the Design of Near-optimal Variable-length Error-correcting Codes for Large-scale Source Alphabets,” IEEE Transactions on Communications, vol. 68, no.12, pp. 7896-7910, Dec. 2020.

[10] H. Hijazi, A. Diallo, M. Kieffer, L. Liberti, and C. Weidmann, “A MILP approach for designing robust variable-length codes based on exact free distance computation,” in Proc. 2012 Data Compression Conference, 10-12 Apr. 2012, Snowbird, USA, pp. 257-266.

[11] Y.-M. Huang, T.-Y. Wu, and Y. S. Han, “An A*-based algorithm for constructing reversible variable-length codes with minimum average codeword length,” IEEE Trans. Commun., vol. 58, no. 11, pp. 3175-3185, Nov. 2010.

[12] T.-Y. Wu, P.-N. Chen, F. Alajaji, and Y. S. Han, “On the design of variable-length error-correcting codes,” IEEE Trans. Commun., vol. 61, no. 9, pp. 3553-3565, Aug. 2013.

[13] K.-C. Lee, S.-H. Wang, C.-P. Li, H.-H. Chang, and H.-J. Li, “Adaptive resource allocation algorithm based on cross-entropy method for OFDMA systems,” IEEE Transactions on Broadcasting, vol. 60, no. 3, pp. 524–531, Sept 2014.

[14] R. Y. Rubinstein, “Optimization of computer simulation models with rare events,” European Journal of Operations Research, 99:89-112, 1997.

[15] R. Y. Rubinstein and D. P. Kroese, “The cross-entropy method,” Berlin, Germany: Springer, 2004.

[16] A. Diallo, C. Weidmann, and M. Kieffer, “Efficient Computation and Optimization of the Free Distance of Variable-Length Finite-State Joint Source-Channel Codes,” IEEE Transactions on Communications, vol. 59, no. 4, pp. 1043-1052, Apr. 2011.