

列向量等权互异且行向量准等权矩阵 的最优生成算法

陈 溧

摘要 本文引入一个递归型矩阵, 给出了 $GF(2)$ 上列向量等权互异且行向量准等权矩阵的一种生成算法, 这个算法是最优时空的, 它可直接用于生成有名的最佳最小奇权列码 (SEC-DED 码) 校验矩阵。最后, 我们把算法推广到 n 元集上。

一、引言

本文给出了一类矩阵的生成算法。这类矩阵有这样的特性: 第一, 矩阵中无相同的列向量; 第二, 对于矩阵中的任一元, 每一列该元的个数相同, 每一行该元的个数之差不超过 1。这样的矩阵称为“列向量等权互异且行向量准等权矩阵”(后文中简称列/行等权矩阵)。

首先, 我们引入了一个递归形式的矩阵, 然后在这个递归矩阵上进行循环操作(计算), 逐步分层、分块地生成列/行等权矩阵。我们的算法是最优时空的。

列/行等权矩阵是一类重要的矩阵, 它可以直接用于生成有名的最佳最小奇权列码校验矩阵(最佳 Hamming 校验矩阵)^{[1][2]}, 并能解决一类作业安排问题。

我们还应用本文的递归矩阵研究了 tEC-MUED 定权码^[3]。

二、(0, 1)型列/行等权矩阵的生成算法

本节将讨论 (0, 1) 型列/行等权矩阵的生成算法, 并把它用于生成最佳最小奇权列码校验矩阵。

符号说明:

$\Delta(R, J, m)$ —— (0, 1) 型 ($R \times m$) 矩阵, 它的列向量的权都为 J , $0 \leq J \leq R$;

$\langle t \rangle(m)$ —— (0, 1) 型 ($1 \times m$) 矩阵, 它的行元素全为 t ;

。——矩阵的上下合并运算符，即

$$N \circ M = \begin{bmatrix} N \\ M \end{bmatrix};$$

⊕——矩阵的左右合并运算符，即 $N \oplus M = [NM]$;

⊖——把该运算符后面的矩阵倒置，然后“⊕”起来，最后把“1”个数多的行变换到矩阵的上部。

定义1 在 $\Delta(R, J, m)$ 中，如果有

$$0 \leq J \leq R \quad \& \quad 0 \leq m \leq C_R^J \quad (1)$$

成立，则称 $\Delta(R, J, m)$ 满足L-条件。

定义2 若满足L-条件的 $\Delta(R, J, m)$ 中任意两行“1”的个数之差不超过1，且无相同的列向量，那么称 $\Delta(R, J, m)$ 为 (0, 1) 型列/行等权矩阵，并可认为空矩阵 (ϕ) 也是列/行等权矩阵。

定义3 如果 $\Delta(R, J, m)$ 满足L-条件，规定：

结果 \ 条件	$m=0$	$J=0$	$J=R$	$m=1$	$J=1$	$J=R-1$
$\Delta(R, J, m) =$	ϕ	$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Bigg\} J$	$\begin{bmatrix} 1 & 0 \\ & \ddots \\ & & 1 \end{bmatrix} \Bigg\} m$	$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ 1 & 0 & & \vdots \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 0 \end{bmatrix}$

显然，表中六类矩阵都是列/行等权矩阵，称它们为结束状态。这里“1”的个数多的行集中在矩阵的上端。

根据定义3，可以假设以后讨论的 $\Delta(R, J, m)$ 满足

$$2 \leq J \leq R-2 \quad (2)$$

显然，满足L-条件的 $\Delta(R, J, m)$ 一定可以表为下面的形式：

$$\Delta(R, J, m) = \begin{bmatrix} 11 \dots \dots \dots 100 \dots \dots \dots 0 \\ \Delta_1(R-1, J-1, m_1) \Delta_2(R-1, J, m-m_1) \end{bmatrix}$$

$$= [\langle 1 \rangle(m_1) \circ \Delta_1(R-1, J-1, m_1)] \oplus [\langle 0 \rangle(m_2) \circ \Delta_2(R-1, J, m-m_1)] \quad (3)$$

定理1 设 $\Delta(R, J, m)$ 满足L-条件 (即 $0 \leq J \leq R-2, 0 \leq m \leq C_R^J$)，令 $m_1 = \lfloor \frac{mJ}{R} + \frac{R-1}{R} \rfloor$ ，则 $\Delta_1(R-1, J-1, m_1)$ 和 $\Delta_2(R-1, J, m-m_1)$ 也满足L-条件。

证 显然有

$$\frac{mJ}{R} \leq m_1 < \frac{mJ}{R} + 1,$$

又由 $m \leq C_R^J$ ，则 $\frac{mJ}{R} \leq C_R^J - 1$ ，于是有

根据我
 $\leq C_R^J$
由
 $\Delta(R, J$
定
 $m_1 =$

也是列/
证

是行向

又因为

所以

于是有

这
1, J,
J, m)
有
长, 这

出
出
J₂,
1, 2,

$$\frac{mJ}{R} \leq m_1 \leq C_{k-1}^{J-1}$$

根据我们的假设(2), 得 $\Delta_1(R-1, J-1, m_1)$ 满足 L -条件。而 $m-m_1 \leq m - \frac{mJ}{R} \leq C_k^J \left(\frac{R-J}{R}\right)$, 所以 $m-m_1 \leq C_{k-1}^J$, 即 $\Delta_2(R-1, J, m-m_1)$ 也满足 L -条件。证毕。

由定理1不难看出, $\Delta(R, J, m)$ 满足 L -条件, 那么它必然可以递归表出, 也就是 $\Delta(R, J, m)$ 是一种递归型矩阵。

定理2 设 $\Delta(R-1, J-1, m_1)$ 和 $\Delta(R-1, J, m-m_1)$ 都是列/行等权矩阵, 且

$$m_1 = \left\lfloor \frac{mJ}{R} + \frac{R-1}{R} \right\rfloor, \text{ 则}$$

$$\Delta'(R, J, m) = [\langle 1 \rangle(m_1) \oplus \langle 0 \rangle(m-m_1)] \circ [\Delta(R-1, J-1, m_1) \oplus \Delta(R-1, J, m-m_1)] \quad (4)$$

也是列/行等权矩阵。

证 显然

$$[\Delta(R-1, J-1, m_1) \oplus \Delta(R-1, J, m-m_1)] \quad (5)$$

是行向量准等权的。式中“1”的总数为 $(mJ-m_1)$, 记它的行权较小者为 d , 有

$$\frac{mJ-m_1}{R-1} - 1 < d \leq \frac{mJ-m_1}{R-1}$$

又因为

$$\frac{mJ}{R} \leq m_1 \leq \frac{mJ}{R} + \frac{R-1}{R}$$

所以

$$\frac{mJ-m_1}{R-1} \leq m_1 \leq \frac{mJ-m_1}{R-1} + 1$$

于是有

$$0 \leq m_1 - d \leq 1$$

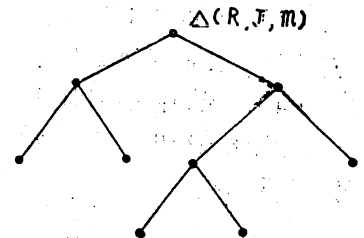
这样, $\Delta'(R, J, m)$ 也是行向量准等权矩阵。又由于 $\Delta(R-1, J-1, m_1)$ 和 $\Delta(R-1, J, m-m_1)$ 中均无相同的列向量, 因此 $\Delta'(R, J, m)$ 也无相同的列向量, 所以 $\Delta'(R, J, m)$ 是列/行等权矩阵。证毕。

有了定理1、2以后, 我们就可以设计列/行等权矩阵的生成算法了。为了避免文章冗长, 这里只给出算法主要的步骤。

算法 A

步骤1 检验 $\Delta(R, J, m)$ 是否满足 L -条件。

步骤2 分解 $\Delta(R, J, m)$ 为 $\Delta(R_1, J_1, m_1), \Delta(R_2, J_2, m_2), \dots, \Delta(R_n, J_n, m_n)$, 这里 $\Delta(R_i, J_i, m_i) (i=1, 2, \dots, n)$ 都是结束状态。



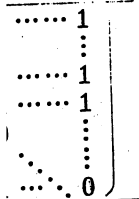
986 年

行变换到

(1)

过1, 且
矩阵(φ)

-1



的个数多的

(2)

$m-m_1$]

(3)

, 令 $m_1 =$

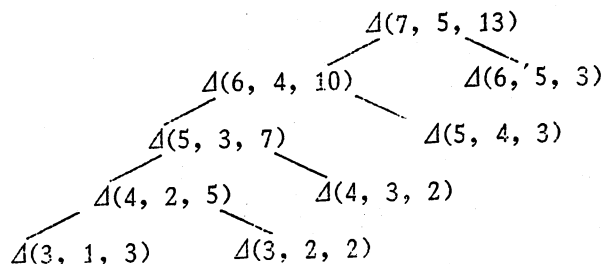
L-条件。

步骤3 依秩把 $\Delta(R_i, J_i, m_i)$ 用 “ \oplus ” 算子合并在一起。注意, “分解”和 “合并” 的对象一致。

步骤2 实际上就是产生了一个二叉树。步骤3就是把某结点的两个 “儿子” 结点 \oplus 合并 (还需根据定理2附加一行)。算法A也可以边 “分解” 边 “合并”。

例: 设 $R=7, J=5, m=13$; 求 $\Delta(R, J, m)$ 。

解: 由算法A中的步骤1, $\Delta(7, 5, 13)$ 检验后确实满足L-条件。再由步骤2, 得



二叉树上的叶子都满足定义3中的条件, 于是由步骤3得

$$\Delta(7, 5, 13) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} .$$

我们容易证明

定理3 算法A的时间复杂性为 $O(R \cdot m)$, 因此, 算法A是最优时间算法。它显然也是最优空间的。

定理4 $\Delta(R, J, m)$ 可以成为列/行等权矩阵的充要条件是 $\Delta(R, J, m)$ 满足L-条件。

上面, 已经给出了GF(2)上列/行等权矩阵的生成算法, 现在讨论它在编码理论上的应用。

“最佳最小奇权列码”是Hsiao(1970)提出的。张焕国(1980)给出了该码校验矩阵的一种生成算法(循环向量法)。

“最佳最小奇权列码”校验矩阵H是这样一种矩阵:

- (1) 每列含奇数个 “1”;
- (2) 总的 “1” 的个数最少;
- (3) 任两行 “1” 的个数之差不超过1;
- (4) 无相同的列向量。

现在就来给出H的生成算法, 这个算法也是最佳的。假如要生成信息位为k的SEC-DED码。首先根据

$$R \geq 1 + \log_2(k+R) \quad (6)$$

来确定 R 。

然后，对于确定的 R ，根据算法 A 求作

$$\Delta(R, 2i+1, C_R^{2i+1}) \quad i=0, 1, \dots, I-1. \quad (7)$$

这里，

$$\sum_{i=0}^{I-1} C_R^{2i+1} \leq k+R \leq \sum_{i=0}^I C_R^{2i+1}. \quad (8)$$

令 $m = (k+R) - \sum_{i=0}^{I-1} C_R^{2i+1}$ ，再由算法 A 生成

$$\Delta(R, 2I+1, m).$$

那末

$$H = \left[\bigoplus_{i=0}^{I-1} \Delta(R, 2i+1, C_R^{2i+1}) \right] \bigoplus \Delta(R, 2I+1, m) \quad (9)$$

就是我们需求的“最佳最小奇权列码”校检矩阵。

三、(1, 2, ..., n)型列/行等权矩阵的生成算法

定义 4 设有矩阵 $M = [m_{ij}]_{R \times m}$ ，其中 $m_{ij} \in S = \{1, 2, \dots, n\}$ 称 M 为 $(1, 2, \dots, n)$ 型矩阵。(后文均假设 M 为 $(1, 2, \dots, n)$ 型矩阵。)

定义 5 对于 M 的某一行 (列) A ， A 中 $(i \in S)$ 的个数称为 i 在 A 中的权。

定义 6 如果对 S 中任一元素 i ， M 的每一列 i 的个数都相同，任两行 i 的个数之差不超过 1，且 M 无相同的列向量，称 M 为列/行等权矩阵。

根据定义 6，不难知道 M 的每一列对应于“不尽相异元素的排列”中的一种取法。我们约定， $M(\{w_1, \dots, w_n\}, m)$ 是 $R = \sum_{i=1}^n w_i$ 行， m 列矩阵，且每一列 i 的个数为 w_i 。显然，如果 $M(\{w_1, \dots, w_n\}, m)$ 是列/行等权的，则必有

$$m \leq R! / \prod_{i=1}^n (w_i!). \quad (10)$$

定理 5 如果 $M(\{w_1, \dots, w_n\}, m)$ 中有 $m = R! / \prod_{i=1}^n (w_i!)$ ，那么令

$$\begin{cases} M(\{w_1, \dots, w_n\}, m) = \bigoplus_{i=1}^n [i] \circ M(\{w_1, \dots, w_{i-1}, \dots, w_n\}, m_i); \\ M(\{0, w_2, \dots, w_n\}, m) = M(\{w_2, \dots, w_n\}, m); \\ M(\{w_i\}, m) = (i, \dots, i)^T, \end{cases} \quad (11)$$

这里

$$m_i = \binom{R-1}{w_1, \dots, w_{i-1}, \dots, w_n},$$

则 $M(\{w_1, \dots, w_n\}, m)$ 是列/行等权的。

上述定理的证明暂略。这个定理给出一种特殊情形的列/行等权矩阵的生成算法。一般

地,可用逐个生成的办法,即首先生成码字“1”,这时把“2”,...,“n”视为同一元——待定元,记为“0”,它的个数 $Q = \sum_{i=2}^n w_i$ 。且令

$$M(\{w_1, w_2, \dots, w_n\}, m) = M(\{w_1; Q\}, m), \tag{12}$$

然后生成码字“2”, ..., “n”。

定义 7 在 $M(\{w_1; Q\}, m)$ 中,如果

$$0 \leq m \leq C_{w_1+Q}^{w_1} (w_2, \dots, w_n) \tag{13}$$

称 $M(\{w_1; Q\}, m)$ 满足 L -条件。

类似于上一节,可以假设

$$w_i \geq w_{i+1}, \quad (i: 2 \leq w_i \leq \sum_{j=1}^i w_j - 2). \tag{14}$$

这时 $M(\{w_1; Q\}, m)$ 可表为

$$M(\{w_1; Q\}, m) = \begin{bmatrix} 1 & 1 & \dots & \dots & \dots & 1 & 0 & 0 & 0 & \dots & \dots & 0 \\ M_1(\{w_1-1; Q\}, m_1) & M_2(\{w_1; Q-1\}, m_2) & & & & & & & & & & \end{bmatrix}. \tag{15}$$

定理 6 设 $M(\{w_1; Q\}, m)$ 满足 L -条件。令 $m_1 = \left[\frac{mw_1}{R} + \frac{R-1}{R} \right] (R = w_1 + Q)$, $m_2 = m - m_1$, 则 $M_1(\{w_1-1; Q\}, m_1)$ 和 $M_2(\{w_1; Q-1\}, m_2)$ 也满足 L -条件。

定理 7 设 $M(\{w_1-1; Q\}, m_1)$ 和 $M(\{w_1; Q-1\}, m-m_1)$ 都是列/行等权的, 则

$$M'(\{w_1; Q\}, m) = [\langle 1 \rangle(m_1) \oplus \langle 0 \rangle(m-m_1)] \circ [M(\{w_1-1; Q\}, m_1) \oplus M(\{w_1; Q-1\}, m-m_1)] \tag{16}$$

也是列行等权的。

定理 6、7 的证明类似于定理 1、2, 这里暂略。注意式 (16) 中可能出现 $r \leq \binom{Q}{w_1, \dots, w_n}$ 个列相同, 这是待定元导致的。把这 r 列选出生成 $M(\{w_2, \dots, w_n\}, r)$ 就是算法的递归过程。同样它是最优时空的。本文不再赘述。

在实际工作中, 有时会遇到这样一类问题: 某车间有 R 台机器 (或工人), I_1, I_2, \dots, I_R ; 加工 n 种另件 $1, 2, \dots, n$, 其中另件 i 每天需要 w_i 个。设每台机器一天只能做一个另件, 即 $R = \sum_{i=1}^n w_i$ 。问怎样安排方能使 m 天内每台机器做另件 $i (i=1, 2, \dots, n)$ 的次数 (几平) 相同? 这类作业安排问题, 用本节的方法就能获得解决。

本文得到张焕国、王伯英老师的指导和帮助, 在此深表感谢。

参 考 文 献

[1] 张焕国; 最佳最小奇重量列码检验矩阵的一种生成算法, 《武汉大学学报》(自

1986年

同一元

(12)

(13)

(14)

(15)

m_2

权的, 则

(16)

(\dots, w_n)
的递归过

$I_2, \dots,$

一个另

数 (几

» (自

然科学版), No.4, 1980.

[2] 陈深: tEC-MUED系统码的一个注记, 《计算机学报》, Vol 7, No.4, 1984.

[3] Hsiao, M. Y.: A Class of Optimal minimum oddweight-column SEC-DED Codes, IBM Reserach and Development, Vol 14, No.4, 1970.

An Optimal Generating Algorithm for Matrix of Equal-Weight Column and Quasiequal-Weight Row

Chen Li

Department of Mathematics and Mechanics

ABSTRACT

According to a recursive matrix which is introduced in this paper, an algorithm for matrix of equal-weight column which is inequal each other and quasiequal-weight row has been given. The algorithm is optimal in time and space, and it can be applied to generate check matrix of "Optimal minimum odd-weight-column SEC-DED codes". Finally, this algorithm has been extended to set of n elements.