



空间填充正交设计的新进展

王春燕^{1,2}, 孙法省^{3*}

1. 中国人民大学应用统计科学研究中心, 北京 100872;

2. 中国人民大学统计学院, 北京 100872;

3. 东北师范大学数学与统计学院, 长春 130024

E-mail: chunyanwang@ruc.edu.cn, sunfs359@nenu.edu.cn

收稿日期: 2024-08-20; 接受日期: 2024-12-14; 网络出版日期: 2025-03-10; * 通信作者

国家自然科学基金 (批准号: 12301323 和 12371259) 和教育部人文社会科学重点研究基地重大项目 (批准号: 22JJD110001) 资助项目

摘要 计算机试验被广泛应用于研究多个领域的复杂系统. 正交性和空间填充性是计算机试验设计的重要性质. 空间填充性确保有效的因子筛选, 而正交性则使得线性模型中主效应的估计互不相关. 本文提出一种构造新型正交设计的通用旋转方法. 该方法操作简便, 在试验次数和水平数上均具灵活性. 所得设计不仅具有正交性, 还具有理想的空间填充性. 这些设计可以是对称的, 也可以是非对称的. 相关的理论支持已得到充分论证. 数值结果也展示了所得设计的优越性.

关键词 计算机试验 分组空间填充设计 列正交性 半旋转矩阵

MSC (2020) 主题分类 62K15, 62K99

1 引言

计算机试验广泛应用于构建复杂仿真系统的高质量替代模型. 空间填充设计在试验区域内均匀分布设计点, 非常适合计算机试验 (参见文献 [4, 19]). 根据效应稀疏性原则^[32], 在试验初期, 试验者通常需要考虑许多因子, 而其中只有少数因子可能是活跃的. 从这个角度来看, 构造具有理想的低维空间填充性的设计更有意义. 文献中已经提出了许多有效的方法来构造这类设计, 例如, 拉丁超立方体设计^[15]、随机正交表^[18]、基于正交表的拉丁超立方体设计^[26]、强正交表^[8], 以及可映射近似正交表^[17]. 关于空间填充设计的更多进展, 可参见文献 [6, 7, 9, 20, 27, 30, 34]. 除了空间填充性外, 正交性也是计算机试验设计的重要性质, 它能使线性模型中主效应的估计互不相关. 同时, 正交性也可以视为构造空间填充设计的一个基石 (参见文献 [2, 25]). 文献中已经提出了许多方法来构造正交设计 (参见文献 [5, 12–14, 22–24, 28, 29, 35, 36, 38]).

空间填充性和正交性是计算机试验设计的理想性质, 可以作为计算机试验设计的评判标准. 文献中已经提出了许多构造空间填充设计和正交设计的方法, 而对于同时具有这两种性质的设计的研究却很少. 相关课题亟需进一步研究.

英文引用格式: Wang C Y, Sun F S. New results for space-filling orthogonal designs (in Chinese). Sci Sin Math, 2026, 56: 789–804, doi: 10.1360/SCM-2024-0514

本文介绍了一类新的空间填充正交设计, 这些设计不仅具有正交性, 而且具有列分组结构, 使得不同组的列具有理想的二维和三维分层性. 此外, 这类设计具有灵活的试验次数和水平数. 对于给定的试验次数, 我们可以获得具有不同水平数的正交设计. 本文提出了构造此类设计的通用方法. 该方法易于实现, 所得设计可以是对称的, 也可以是非对称的. 相关的理论支持已得到充分论证. 数值结果也展示了所得设计的优越性.

本文余下内容的结构如下. 第 2 节介绍相关符号和预备知识. 第 3 节提出一种构造空间填充正交设计的通用方法. 第 4 节针对因子具有不同水平数的试验, 考虑混合水平空间填充正交设计. 第 5 节通过一个应用实例来展示所得设计在构建统计替代模型方面的有效性. 第 6 节总结全文. 附录给出定理的证明.

2 符号与预备知识

用 $D(n, s_1, \dots, s_m)$ 表示一个包含 n 次试验、 m 个因子和 $s_1, \dots, s_m$ 个水平的混合水平设计. 在这个设计中, 对于 $i = 1, \dots, m$, 第 i 列的 s_i 个水平来自 $\{0, 1, \dots, s_i - 1\}$, 并且每个水平均以相同频率出现. 当所有 s_j 都等于 s 时, 该设计变为一个对称设计, 记作 $D(n, s^m)$. 两个列向量 $a = (a_1, \dots, a_n)^T$ 和 $b = (b_1, \dots, b_n)^T$ 之间的相关系数定义为

$$\rho(a, b) = \sum_{i=1}^n (a_i - \bar{a})(b_i - \bar{b}) / \left[\sum_{i=1}^n (a_i - \bar{a})^2 \sum_{i=1}^n (b_i - \bar{b})^2 \right]^{1/2},$$

其中

$$\bar{a} = \sum_{i=1}^n a_i / n, \quad \bar{b} = \sum_{i=1}^n b_i / n.$$

如果设计 $D(n, s^m)$ 中任意两列的相关系数都为 0, 则该设计为正交设计, 记作 $OD(n, s^m)$. 为研究正交性, 有时将设计 D 的 s 个水平中心化到 $\Omega(s) = \{z - (s-1)/2 \mid z = 0, \dots, s-1\}$ 上, 并将得到的设计记为 $\tilde{D}$.

若设计 $D(n, s_1, \dots, s_m)$ 中任意 t 列的所有可能水平组合均以相同频率出现, 则该设计是强度为 t 的正交表, 记作 $OA(n, m, s_1 \times \dots \times s_m, t)$. 当所有 s_j 都等于 s 时, 该正交表是对称的, 记作 $OA(n, m, s, t)$.

用 $GF(s)$ 表示 s 阶 Galois 域. 对于 $GF(s)$ 上的一个 $q \times c$ 矩阵, 如果对于任意的 $1 \leq i < j \leq c$, 第 i 和 j 列之间的向量差以相等的频率包含 $GF(s)$ 中的每个元素, 则称该矩阵为 $GF(s)$ 上的差阵, 记作 $D(q, c, s)$. 对于 $GF(s)$ 上的两个矩阵 $A = (a_{ij})_{m \times n}$ 和 $B = (b_{ij})_{u \times v}$, 它们的 Kronecker 和定义为

$$A \oplus B = \begin{bmatrix} a_{11} \dot{+} B & \dots & a_{1n} \dot{+} B \\ \vdots & \ddots & \vdots \\ a_{m1} \dot{+} B & \dots & a_{mn} \dot{+} B \end{bmatrix},$$

其中 $\dot{+}$ 是定义在 $GF(s)$ 上的加法.

对于一个素数或素数幂 s , 令

$$R_0 = \begin{bmatrix} s^2 & 0 & 1 & s \\ 1 & s & -s^2 & 0 \end{bmatrix} \quad \text{和} \quad Q_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

为两个大小为 2×4 的半旋转矩阵, 并定义

$$R_k = \begin{bmatrix} sR_{k-1} & -Q_{k-1} \\ Q_{k-1} & sR_{k-1} \end{bmatrix}, \quad \text{其中 } Q_k = \begin{bmatrix} Q_{k-1} & 0 \\ 0 & -Q_{k-1} \end{bmatrix}, \quad k \geq 1,$$

则基于 R_0 可获得一系列半旋转矩阵 $R_1, R_2, R_3, \dots$, 其中 R_k 的大小为 $2^{k+1} \times 2^{k+2}$. 可以验证, 对于 $k \geq 0$, R_k 的行相互正交, 即 $R_k R_k^T = \eta_k I_{2^{k+1}}$, 其中 $\eta_k = 21 + 64(4^k - 1)/3$, $I_{2^{k+1}}$ 是阶数为 2^{k+1} 的单位矩阵.

3 对称空间填充正交设计

本节介绍一种对称空间填充正交设计的构造方法, 并研究所得设计的性质. 具体的构造过程见算法 1. 根据定理 3.1, 所得到的设计不仅具有正交性, 还具有理想的分层性.

算法 1 对称空间填充正交设计 $\text{OD}(s^{k+1}n, (s^{k+3})^{gc})$ 的构造

输入: 一个素数或素数幂 s ; 一个 $\text{OA}(n, g, p, 2)$, 记作 A ; 一个 $\text{OA}(p, f, s, 2)$, 记作 B ; 一个差阵 $D(s, s, s)$, 记作 $D = (d_0, \dots, d_{s-1})$; 一个非负整数 k .

输出: 一个 $\text{OD}(s^{k+1}n, (s^{k+3})^{gc})$, 记作 W , 其中 $c = \lambda 2^{k+1} t^k$, $\lambda = \lfloor f/2 \rfloor$, $t = \lfloor s/2 \rfloor$.

步骤 1: 在 A 的每一列中, 将水平 $v-1$ 替换为 B 的第 v 行 ($v = 1, \dots, p$), 并得到 $C = (C_1, \dots, C_g)$, 其中 C_i 表示通过将 A 的第 i 列的水平替换为 B 的行而获得的 f 个列. 对于 $i = 1, \dots, g$, 将每个 C_i 划分为 λ 或 $\lambda+1$ 个组, 并得到 $C_i = (C_{i,1}, \dots, C_{i,\lambda})$, 若 $f = 2\lambda$; 或 $C_i = (C_{i,1}, \dots, C_{i,\lambda}, t_i)$, 若 $f = 2\lambda+1$, 其中 $\lambda = \lfloor f/2 \rfloor$, 每个 $C_{i,j}$ 包含两列.

步骤 2: 对于 $i = 1, 2, \dots, g$, $j = 1, 2, \dots, \lambda$, 将每个 $C_{i,j}$ 中的所有元素视为 $GF(s)$ 中的元素, 并令

$$C_{i,j}^{r_1} = (d_{2r_1} \oplus C_{i,j}, d_{2r_1+1} \oplus C_{i,j}), \quad r_1 = 0, 1, \dots, t-1,$$

可得 t 个大小为 $sn \times 4$ 的设计 $C_{i,j}^0, \dots, C_{i,j}^{t-1}$. 一般地, 令

$$C_{i,j}^{r_1 r_2 \dots r_k} = (d_{2r_k} \oplus C_{i,j}^{r_1 r_2 \dots r_{k-1}}, d_{2r_k+1} \oplus C_{i,j}^{r_1 r_2 \dots r_{k-1}}), \quad r_k = 0, 1, \dots, t-1,$$

则可得 t^k 个大小为 $s^k n \times 2^{k+1}$ 的设计 $C_{i,j}^{00\dots 0}, \dots, C_{i,j}^{(t-1)(t-1)\dots(t-1)}$.

步骤 3: 对于 $r_1, r_2, \dots, r_k = 0, 1, \dots, t-1$, 用 c_l 表示 $C_{i,j}^{r_1 r_2 \dots r_k}$ 的第 l 个列, 并令

$$E_{i,j}^{r_1 r_2 \dots r_k} = (d_0 \oplus c_1, d_1 \oplus c_1, \dots, d_0 \oplus c_{2k+1}, d_1 \oplus c_{2k+1}).$$

将 $E_{i,j}^{r_1 r_2 \dots r_k}$ 的每一列的 s 个水平中心化, 并将得到的设计记作 $\tilde{E}_{i,j}^{r_1 r_2 \dots r_k}$. 通过半旋转矩阵 R_k 对每个 $\tilde{E}_{i,j}^{r_1 r_2 \dots r_k}$ 进行旋转, 并定义

$$W_{i,j} = (\tilde{E}_{i,j}^{00\dots 0} R_k^T, \tilde{E}_{i,j}^{00\dots 1} R_k^T, \dots, \tilde{E}_{i,j}^{(t-1)(t-1)\dots(t-1)} R_k^T), \quad i = 1, 2, \dots, g, \quad j = 1, 2, \dots, \lambda.$$

通过列并置将 $W_{i,1}, \dots, W_{i,\lambda}$ 组合在一起得到 $W_i = (W_{i,1}, W_{i,2}, \dots, W_{i,\lambda})$, 并定义

$$W = (W_1, W_2, \dots, W_g).$$

定理 3.1 由算法 1 生成的设计 W 是一个 $\text{OD}(s^{k+1}n, (s^{k+3})^{gc})$, 其中 $c = \lambda 2^{k+1} t^k$, $\lambda = \lfloor f/2 \rfloor$, $t = \lfloor s/2 \rfloor$. W 可以划分为 g 组, 每组包含 c 列, 并具有以下性质:

- (i) 任意两列实现 $s \times s^2$ 和 $s^2 \times s$ 的格子分层;
- (ii) 来自不同组的两列实现 $s \times s^3$, $s^2 \times s^2$ 和 $s^3 \times s$ 的格子分层;
- (iii) 来自两个组的三列实现 $s \times s \times s$ 的格子分层.

根据定理 3.1, 对于设计 W , 实现 $s \times s^3$, $s^2 \times s^2$ 和 $s^2 \times s$ 格子分层的二维子矩阵的比例至少为 $\pi = (g-1)c/(gc-1)$.

例 3.1 设 A 是一个 $OA(16, 5, 4, 2)$, B 是一个 $OA(4, 3, 2, 2)$, D 是一个 $D(2, 2, 2)$, 如下所示:

$$A^T = \begin{pmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 0 & 3 & 2 & 2 & 3 & 0 & 1 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 2 & 3 & 0 & 1 & 3 & 2 & 1 & 0 & 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 & 3 & 2 & 1 & 0 & 1 & 0 & 3 & 2 & 2 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$

这里 $n = 16$, $g = 5$, $p = 4$, $f = 3$, $s = 2$, $\lambda = \lfloor f/2 \rfloor = 1$, $t = \lfloor s/2 \rfloor = 1$. 对于 $i = 1, \dots, 5$, 将 A 的第 i 列的水平 $v-1$ 替换为 B 的第 v 行 ($v = 1, \dots, 4$), 并将生成的设计记作 C_i . 设 c_{il} 为 C_i 的第 l 列, 定义 $E_i = (d_0 \oplus c_{i1}, d_1 \oplus c_{i1}, d_0 \oplus c_{i2}, d_1 \oplus c_{i2})$. 将 E_i 的两个水平中心化, 并将得到的设计记作 $\tilde{E}_i$. 通过半旋转矩阵 R_0 对 $\tilde{E}_i$ 进行旋转, 得到 $W_i = \tilde{E}_i R_0^T$. 通过列并置将 $W_1, \dots, W_5$ 组合在一起, 得到一个 $OD(32, 8^{10})$, 如表 1 所示. 容易验证, W 可以划分为五组, 每组包含两列, 使得任意两列都可以塌陷为一个 $OA(32, 2, 2 \times 4, 2)$ 和一个 $OA(32, 2, 4 \times 2, 2)$, 如图 1 所示. 来自不同组的任意两列都可以塌陷为一个 $OA(32, 2, 2 \times 8, 2)$ 、一个 $OA(32, 2, 4, 2)$ 和一个 $OA(32, 2, 8 \times 2, 2)$, 如图 2 所示. 因此, W 在所有 45 个二维子矩阵中的 40 个上实现了 2×8 , 4×4 和 8×2 的格子分层, 因此 $\pi = 88.89\%$. 此外, 考虑

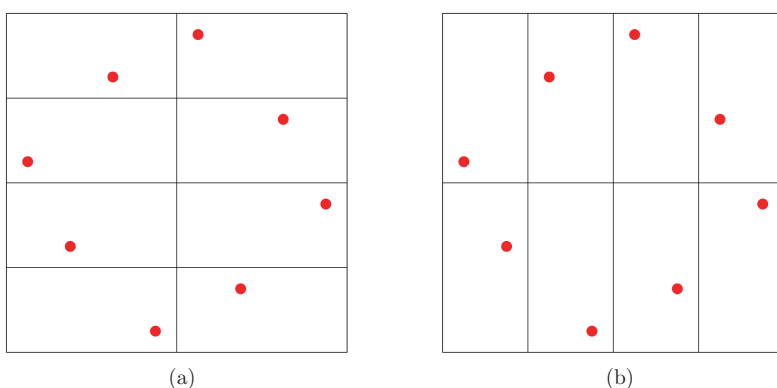


图 1 (网络版彩图) 例 3.1 中设计 $OD(32, 8^{10})$ 的前两列的分层性. (a) 2×4 的格子分层; (b) 4×2 的格子分层

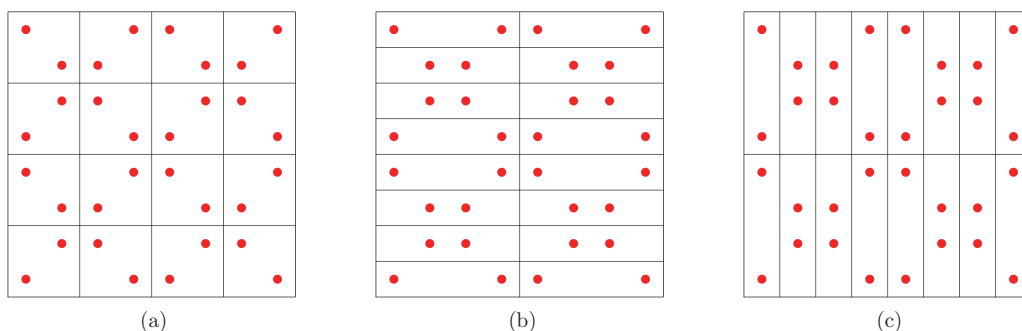


图 2 (网络版彩图) 例 3.1 中设计 $OD(32, 8^{10})$ 的第一、三列的分层性. (a) 4×4 的格子分层; (b) 2×8 的格子分层; (c) 8×2 的格子分层

表 1 例 3.1 中的 $\text{OD}(32, 8^{10})$, $W = (W_1, W_2, W_3, W_4, W_5)$

$W_1 = \tilde{E}_1 R_0^T$		$W_2 = \tilde{E}_2 R_0^T$		$W_3 = \tilde{E}_3 R_0^T$		$W_4 = \tilde{E}_4 R_0^T$		$W_5 = \tilde{E}_5 R_0^T$	
-3.5	0.5	-3.5	0.5	-3.5	0.5	-3.5	0.5	-3.5	0.5
-0.5	-3.5	-0.5	-3.5	-0.5	-3.5	-0.5	-3.5	-3.5	0.5
-3.5	0.5	-0.5	-3.5	0.5	3.5	3.5	-0.5	-0.5	-3.5
-0.5	-3.5	-3.5	0.5	3.5	-0.5	0.5	3.5	-0.5	-3.5
-3.5	0.5	0.5	3.5	3.5	-0.5	-0.5	-3.5	0.5	3.5
-0.5	-3.5	3.5	-0.5	0.5	3.5	-3.5	0.5	0.5	3.5
-3.5	0.5	3.5	-0.5	-0.5	-3.5	0.5	3.5	3.5	-0.5
-0.5	-3.5	0.5	3.5	-3.5	0.5	3.5	-0.5	3.5	-0.5
-1.5	2.5	-1.5	2.5	-1.5	2.5	-1.5	2.5	-1.5	2.5
-2.5	-1.5	-2.5	-1.5	-2.5	-1.5	-2.5	-1.5	-1.5	2.5
-1.5	2.5	-2.5	-1.5	2.5	1.5	1.5	-2.5	-2.5	-1.5
-2.5	-1.5	-1.5	2.5	1.5	-2.5	2.5	1.5	-2.5	-1.5
-1.5	2.5	2.5	1.5	1.5	-2.5	-2.5	-1.5	2.5	1.5
-2.5	-1.5	1.5	-2.5	2.5	1.5	-1.5	2.5	2.5	1.5
-1.5	2.5	1.5	-2.5	-2.5	-1.5	2.5	1.5	1.5	-2.5
-2.5	-1.5	2.5	1.5	-1.5	2.5	1.5	-2.5	1.5	-2.5
3.5	-0.5	3.5	-0.5	3.5	-0.5	3.5	-0.5	-3.5	0.5
0.5	3.5	0.5	3.5	0.5	3.5	0.5	3.5	-3.5	0.5
3.5	-0.5	0.5	3.5	-0.5	-3.5	-3.5	0.5	-0.5	-3.5
0.5	3.5	3.5	-0.5	-3.5	0.5	-0.5	-3.5	-0.5	-3.5
3.5	-0.5	-0.5	-3.5	-3.5	0.5	0.5	3.5	0.5	3.5
0.5	3.5	-3.5	0.5	-0.5	-3.5	3.5	-0.5	0.5	3.5
3.5	-0.5	-3.5	0.5	0.5	3.5	-0.5	-3.5	3.5	-0.5
0.5	3.5	-0.5	-3.5	3.5	-0.5	-3.5	0.5	3.5	-0.5
1.5	-2.5	1.5	-2.5	1.5	-2.5	1.5	-2.5	-1.5	2.5
2.5	1.5	2.5	1.5	2.5	1.5	2.5	1.5	-1.5	2.5
1.5	-2.5	2.5	1.5	-2.5	-1.5	-1.5	2.5	-2.5	-1.5
2.5	1.5	1.5	-2.5	-1.5	2.5	-2.5	-1.5	-2.5	-1.5
1.5	-2.5	-2.5	-1.5	-1.5	2.5	2.5	1.5	2.5	1.5
2.5	1.5	-1.5	2.5	-2.5	-1.5	1.5	-2.5	2.5	1.5
1.5	-2.5	-1.5	2.5	2.5	1.5	-2.5	-1.5	1.5	-2.5
2.5	1.5	-2.5	-1.5	1.5	-2.5	-1.5	2.5	1.5	-2.5

设计 W 的三维分层. 可以验证, 来自两个组的任意三列实现 $2 \times 2 \times 2$ 的格子分层. 事实上, W 中实现 $2 \times 2 \times 2$ 的格子分层的三维子矩阵的比例往往非常高. 全面检查表明, W 在所有 120 个三维子矩阵中的 112 个上实现了 $2 \times 2 \times 2$ 的格子分层 (即 93.33%).

对 $i = 1, \dots, 5$, 设 e_{il} 为 E_i 的第 l 列, 定义 $E_i^0 = (d_0 \oplus e_{i1}, d_1 \oplus e_{i1}, d_0 \oplus e_{i2}, d_1 \oplus e_{i2}, d_0 \oplus e_{i3}, d_1 \oplus e_{i3}, d_0 \oplus e_{i4}, d_1 \oplus e_{i4})$. 将 E_i^0 的两个水平中心化, 并将得到的设计记作 $\tilde{E}_i^0$. 通过半旋转矩阵 R_1 对 $\tilde{E}_i^0$ 进行旋转, 得到 $\tilde{E}_i^0 R_1^T$. 通过列并置将 $\tilde{E}_1^0 R_1^T, \dots, \tilde{E}_5^0 R_1^T$ 组合在一起, 得到一个 $\text{OD}(64, 16^{20})$, 见表 2. 可以验

表 2 例 3.1 中的 $OD(64, 16^{20})$, $W = (W_1, W_2, W_3, W_4, W_5)$

$W_1 = \tilde{E}_1^0 R_1^T$				$W_2 = \tilde{E}_2^0 R_1^T$				$W_3 = \tilde{E}_3^0 R_1^T$				$W_4 = \tilde{E}_4^0 R_1^T$				$W_5 = \tilde{E}_5^0 R_1^T$			
-7.5	0.5	-6.5	1.5	-7.5	0.5	-6.5	1.5	-7.5	0.5	-6.5	1.5	-7.5	0.5	-6.5	1.5	-7.5	0.5	-6.5	1.5
-1.5	-6.5	-0.5	-7.5	-1.5	-6.5	-0.5	-7.5	-1.5	-6.5	-0.5	-7.5	-1.5	-6.5	-0.5	-7.5	-7.5	0.5	-6.5	1.5
-7.5	0.5	-6.5	1.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	0.5	7.5	7.5	-0.5	6.5	-1.5	-1.5	-6.5	-0.5	-7.5
-1.5	-6.5	-0.5	-7.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	1.5	6.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5
-7.5	0.5	-6.5	1.5	1.5	6.5	0.5	7.5	7.5	-0.5	6.5	-1.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	0.5	7.5
-1.5	-6.5	-0.5	-7.5	7.5	-0.5	6.5	-1.5	1.5	6.5	0.5	7.5	-7.5	0.5	-6.5	1.5	1.5	6.5	0.5	7.5
-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	0.5	7.5	7.5	-0.5	6.5	-1.5
-1.5	-6.5	-0.5	-7.5	1.5	6.5	0.5	7.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	7.5	-0.5	6.5	-1.5
-6.5	1.5	7.5	-0.5	-6.5	1.5	7.5	-0.5	-6.5	1.5	7.5	-0.5	-6.5	1.5	7.5	-0.5	-6.5	1.5	7.5	-0.5
-0.5	-7.5	1.5	6.5	-0.5	-7.5	1.5	6.5	-0.5	-7.5	1.5	6.5	-0.5	-7.5	1.5	6.5	-6.5	1.5	7.5	-0.5
-6.5	1.5	7.5	-0.5	-0.5	-7.5	1.5	6.5	0.5	7.5	-1.5	-6.5	6.5	-1.5	-7.5	0.5	-0.5	-7.5	1.5	6.5
-0.5	-7.5	1.5	6.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5
-6.5	1.5	7.5	-0.5	0.5	7.5	-1.5	-6.5	6.5	-1.5	-7.5	0.5	-0.5	-7.5	1.5	6.5	0.5	7.5	-1.5	-6.5
-0.5	-7.5	1.5	6.5	6.5	-1.5	-7.5	0.5	0.5	7.5	-1.5	-6.5	-6.5	1.5	7.5	-0.5	0.5	7.5	-1.5	-6.5
-6.5	1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	-0.5	-7.5	1.5	6.5	0.5	7.5	-1.5	-6.5	6.5	-1.5	-7.5	0.5
-0.5	-7.5	1.5	6.5	0.5	7.5	-1.5	-6.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	6.5	-1.5	-7.5	0.5
-3.5	4.5	-2.5	5.5	-3.5	4.5	-2.5	5.5	-3.5	4.5	-2.5	5.5	-3.5	4.5	-2.5	5.5	-3.5	4.5	-2.5	5.5
-5.5	-2.5	-4.5	-3.5	-5.5	-2.5	-4.5	-3.5	-5.5	-2.5	-4.5	-3.5	-5.5	-2.5	-4.5	-3.5	-3.5	4.5	-2.5	5.5
-3.5	4.5	-2.5	5.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	4.5	3.5	3.5	-4.5	2.5	-5.5	-5.5	-2.5	-4.5	-3.5
-5.5	-2.5	-4.5	-3.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	5.5	2.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5
-3.5	4.5	-2.5	5.5	5.5	2.5	4.5	3.5	3.5	-4.5	2.5	-5.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	4.5	3.5
-5.5	-2.5	-4.5	-3.5	3.5	-4.5	2.5	-5.5	5.5	2.5	4.5	3.5	-3.5	4.5	-2.5	5.5	5.5	2.5	4.5	3.5
-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	4.5	3.5	3.5	-4.5	2.5	-5.5
-5.5	-2.5	-4.5	-3.5	5.5	2.5	4.5	3.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	3.5	-4.5	2.5	-5.5
-2.5	5.5	3.5	-4.5	-2.5	5.5	3.5	-4.5	-2.5	5.5	3.5	-4.5	-2.5	5.5	3.5	-4.5	-2.5	5.5	3.5	-4.5
-4.5	-3.5	5.5	2.5	-4.5	-3.5	5.5	2.5	-4.5	-3.5	5.5	2.5	-4.5	-3.5	5.5	2.5	-2.5	5.5	3.5	-4.5
-2.5	5.5	3.5	-4.5	-4.5	-3.5	5.5	2.5	4.5	3.5	-5.5	-2.5	2.5	-5.5	-3.5	4.5	-4.5	-3.5	5.5	2.5
-4.5	-3.5	5.5	2.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5
-2.5	5.5	3.5	-4.5	4.5	3.5	-5.5	-2.5	2.5	-5.5	-3.5	4.5	-4.5	-3.5	5.5	2.5	4.5	3.5	-5.5	-2.5
-4.5	-3.5	5.5	2.5	2.5	-5.5	-3.5	4.5	4.5	3.5	-5.5	-2.5	-2.5	5.5	3.5	-4.5	4.5	3.5	-5.5	-2.5
-2.5	5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	-4.5	-3.5	5.5	2.5	4.5	3.5	-5.5	-2.5	2.5	-5.5	-3.5	4.5
-4.5	-3.5	5.5	2.5	4.5	3.5	-5.5	-2.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	2.5	-5.5	-3.5	4.5
7.5	-0.5	6.5	-1.5	7.5	-0.5	6.5	-1.5	7.5	-0.5	6.5	-1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5
1.5	6.5	0.5	7.5	1.5	6.5	0.5	7.5	1.5	6.5	0.5	7.5	1.5	6.5	0.5	7.5	-7.5	0.5	-6.5	1.5
7.5	-0.5	6.5	-1.5	1.5	6.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	-7.5	0.5	-6.5	1.5	-1.5	-6.5	-0.5	-7.5
1.5	6.5	0.5	7.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	-1.5	-6.5	-0.5	-7.5	-1.5	-6.5	-0.5	-7.5
7.5	-0.5	6.5	-1.5	-1.5	-6.5	-0.5	-7.5	-7.5	0.5	-6.5	1.5	1.5	6.5	0.5	7.5	1.5	6.5	0.5	7.5
1.5	6.5	0.5	7.5	-7.5	0.5	-6.5	1.5	-1.5	-6.5	-0.5	-7.5	7.5	-0.5	6.5	-1.5	1.5	6.5	0.5	7.5

(续表)

$W_1 = \tilde{E}_1^0 R_1^T$				$W_2 = \tilde{E}_2^0 R_1^T$				$W_3 = \tilde{E}_3^0 R_1^T$				$W_4 = \tilde{E}_4^0 R_1^T$				$W_5 = \tilde{E}_5^0 R_1^T$			
7.5	-0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	1.5	6.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	7.5	-0.5	6.5	-1.5
1.5	6.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5
6.5	-1.5	-7.5	0.5	6.5	-1.5	-7.5	0.5	6.5	-1.5	-7.5	0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5
0.5	7.5	-1.5	-6.5	0.5	7.5	-1.5	-6.5	0.5	7.5	-1.5	-6.5	0.5	7.5	-1.5	-6.5	-6.5	1.5	7.5	-0.5
6.5	-1.5	-7.5	0.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	-6.5	1.5	7.5	-0.5	-0.5	-7.5	1.5	6.5
0.5	7.5	-1.5	-6.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	-0.5	-7.5	1.5	6.5	-0.5	-7.5	1.5	6.5
6.5	-1.5	-7.5	0.5	-0.5	-7.5	1.5	6.5	-6.5	1.5	7.5	-0.5	0.5	7.5	-1.5	-6.5	0.5	7.5	-1.5	-6.5
0.5	7.5	-1.5	-6.5	-6.5	1.5	7.5	-0.5	-0.5	-7.5	1.5	6.5	6.5	-1.5	-7.5	0.5	0.5	7.5	-1.5	-6.5
6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	6.5	-1.5	-7.5	0.5
0.5	7.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5
3.5	-4.5	2.5	-5.5	3.5	-4.5	2.5	-5.5	3.5	-4.5	2.5	-5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5
5.5	2.5	4.5	3.5	5.5	2.5	4.5	3.5	5.5	2.5	4.5	3.5	5.5	2.5	4.5	3.5	-3.5	4.5	-2.5	5.5
3.5	-4.5	2.5	-5.5	5.5	2.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	-3.5	4.5	-2.5	5.5	-5.5	-2.5	-4.5	-3.5
5.5	2.5	4.5	3.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	-5.5	-2.5	-4.5	-3.5	-5.5	-2.5	-4.5	-3.5
3.5	-4.5	2.5	-5.5	-5.5	-2.5	-4.5	-3.5	-3.5	4.5	-2.5	5.5	5.5	2.5	4.5	3.5	5.5	2.5	4.5	3.5
5.5	2.5	4.5	3.5	-3.5	4.5	-2.5	5.5	-5.5	-2.5	-4.5	-3.5	3.5	-4.5	2.5	-5.5	5.5	2.5	4.5	3.5
3.5	-4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	5.5	2.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	3.5	-4.5	2.5	-5.5
5.5	2.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5
2.5	-5.5	-3.5	4.5	2.5	-5.5	-3.5	4.5	2.5	-5.5	-3.5	4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5
4.5	3.5	-5.5	-2.5	4.5	3.5	-5.5	-2.5	4.5	3.5	-5.5	-2.5	4.5	3.5	-5.5	-2.5	-2.5	5.5	3.5	-4.5
2.5	-5.5	-3.5	4.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	-2.5	5.5	3.5	-4.5	-4.5	-3.5	5.5	2.5
4.5	3.5	-5.5	-2.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	-4.5	-3.5	5.5	2.5	-4.5	-3.5	5.5	2.5
2.5	-5.5	-3.5	4.5	-4.5	-3.5	5.5	2.5	-2.5	5.5	3.5	-4.5	4.5	3.5	-5.5	-2.5	4.5	3.5	-5.5	-2.5
4.5	3.5	-5.5	-2.5	-2.5	5.5	3.5	-4.5	-4.5	-3.5	5.5	2.5	2.5	-5.5	-3.5	4.5	4.5	3.5	-5.5	-2.5
2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	2.5	-5.5	-3.5	4.5
4.5	3.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5

证, 这个 $OD(64, 16^{20})$ 可以划分为五组, 每组包含四列, 使得任意两列都可以塌陷为一个 $OA(64, 2, 2 \times 4, 2)$ 和一个 $OA(64, 2, 4 \times 2, 2)$, 来自不同组的任意两列都可以塌陷为一个 $OA(64, 2, 2 \times 8, 2)$ 、一个 $OA(64, 2, 4, 2)$ 和一个 $OA(64, 2, 8 \times 2, 2)$. 该设计在所有 190 个二维子矩阵中的 160 个上实现了 2×8 , 4×4 和 8×2 的格子分层 (84.21%), 并在所有 1,140 个三维子矩阵中的 1,108 个上实现了 $2 \times 2 \times 2$ 的格子分层 (97.19%).

对于算法 1 生成的设计 W , 来自不同组的任意两列实现 $s \times s^3$, $s^2 \times s^2$ 和 $s^3 \times s$ 的格子分层. 值得注意的是, 当 B 取某些特定的正交表时, 所得的正交设计具有更好的二维空间填充性.

定理 3.2 设 A 为一个 $OA(n, g, p, 2)$, B 为一个 $OA(p, f, s, 2)$, 其中 $\lfloor f/2 \rfloor = 1$, $\lfloor s/2 \rfloor = 1$. 则通过算法 1 得到的设计 W 是一个 $OD(s^{k+1}n, (s^{k+3})^{gc})$, 其中 $c = 2^{k+1}$. W 可以划分为 g 组, 每组包含 c 列, 并具有定理 3.1 中所述的性质. 此外, 每个组都可以进一步划分为 2^k 个类, 每类包含两列, 使得来自不同类的任意两列实现 $s \times s^3$ 和 $s^3 \times s$ 的格子分层.

例 3.2 (例 3.1 续) 如例 3.1 所述, 由算法 1 构造的 $OD(64, 16^{20})$ 可以划分为五组, 每组包含四

列, 使得来自不同组的任意两列实现 2×8 , 4×4 和 8×2 的格子分层. 现在, 将每组中的四列进一步划分为两类, 每类两列. 可以验证, 来自不同类的任意两列实现 2×8 和 8×2 的格子分层. 因此, 该设计在所有 190 个二维子矩阵中的 180 个上实现了 2×8 和 8×2 的格子分层 (94.74%).

算法 1 的构造中使用的正交表和差阵可参见网站 <http://neilsloane.com/oadir/index.html> 和文献 [10]. 表 3 总结了由算法 1 生成的正交设计. 这些设计的空间填充性由参数 g (组数)、 c (每组中的列数) 和 π (实现 $s \times s^3$, $s^2 \times s^2$ 和 $s^3 \times s$ 格子分层的二维子矩阵的比例) 来表征. 如表 3 所示, π 的

表 3 算法 1 生成的对称空间填充正交设计

$A: \text{OA}(n, g, p, 2)$	$B: \text{OA}(p, f, s, 2)$	k	$W: \text{OD}(s^{k+1}n, (s^{k+3})^{gc})$	g	c	π (%)
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	0	OD(32, 8^{10}) [†]	5	2	88.89
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	1	OD(64, 16^{20}) [†]	5	4	84.21
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	OD(128, 32^{40}) [†]	5	8	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	3	OD(256, 64^{80}) [†]	5	16	81.01
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	4	OD(512, 128^{160}) [†]	5	32	80.50
OA(32, 9, 4, 2)	OA(4, 3, 2, 2)	0	OD(64, 8^{18}) [†]	9	2	94.12
OA(32, 9, 4, 2)	OA(4, 3, 2, 2)	1	OD(128, 16^{36}) [†]	9	4	91.43
OA(32, 9, 4, 2)	OA(4, 3, 2, 2)	2	OD(256, 32^{72}) [†]	9	8	90.14
OA(32, 9, 4, 2)	OA(4, 3, 2, 2)	3	OD(512, 64^{144}) [†]	9	16	89.51
OA(32, 9, 4, 2)	OA(4, 3, 2, 2)	4	OD(1024, 128^{288}) [†]	9	32	89.20
OA(64, 21, 4, 2)	OA(4, 3, 2, 2)	0	OD(128, 8^{42})	21	2	97.56
OA(64, 21, 4, 2)	OA(4, 3, 2, 2)	1	OD(256, 16^{84})	21	4	96.39
OA(64, 21, 4, 2)	OA(4, 3, 2, 2)	2	OD(512, 32^{168})	21	8	95.81
OA(64, 21, 4, 2)	OA(4, 3, 2, 2)	3	OD(1024, 64^{336})	21	16	95.52
OA(64, 21, 4, 2)	OA(4, 3, 2, 2)	4	OD(2048, 128^{672})	21	32	95.38
OA(128, 41, 4, 2)	OA(4, 3, 2, 2)	0	OD(256, 8^{82})	41	2	98.77
OA(128, 41, 4, 2)	OA(4, 3, 2, 2)	1	OD(512, 16^{164})	41	4	98.16
OA(128, 41, 4, 2)	OA(4, 3, 2, 2)	2	OD(1024, 32^{328})	41	8	97.86
OA(128, 41, 4, 2)	OA(4, 3, 2, 2)	3	OD(2048, 64^{656})	41	16	97.71
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	0	OD(243, 27^{40})	10	4	92.31
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	OD(729, 81^{80})	10	8	91.14
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	OD(2187, 243^{160})	10	16	90.57
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	3	OD(6561, 729^{320})	10	32	90.28
OA(162, 19, 9, 2)	OA(9, 4, 3, 2)	0	OD(486, 27^{76})	19	4	96.00
OA(162, 19, 9, 2)	OA(9, 4, 3, 2)	1	OD(1458, 81^{152})	19	8	95.36
OA(162, 19, 9, 2)	OA(9, 4, 3, 2)	2	OD(4374, 243^{304})	19	16	95.05
OA(256, 17, 16, 2)	OA(16, 5, 4, 2)	0	OD(1024, 64^{68})	17	4	95.52
OA(256, 17, 16, 2)	OA(16, 5, 4, 2)	1	OD(4096, 256^{272})	17	16	94.46
OA(512, 33, 16, 2)	OA(16, 5, 4, 2)	0	OD(2048, 64^{132})	33	4	97.71
OA(512, 33, 16, 2)	OA(16, 5, 4, 2)	1	OD(8192, 256^{528})	33	16	97.15
OA(625, 26, 5, 2)	OA(25, 6, 5, 2)	0	OD(3125, 125^{156})	26	6	96.77
OA(625, 26, 5, 2)	OA(25, 6, 5, 2)	1	OD(15625, 625^{624})	26	24	96.31

值非常接近 1, 表明这些正交设计具有理想的二维分层性.

4 混合水平空间填充正交设计

第 3 节构造了具有 s^{k+3} 个水平的对称空间填充正交设计. 接下来, 考虑这种设计的混合水平情况, 当试验的因素水平数不同时, 这种设计非常有用. 具体的构造过程见算法 2.

算法 2 混合水平空间填充正交设计 $\text{OD}(s^{r+1}n, \Pi_{l=0}^r(s^{r+3-l})_{g_l c_l})$ 的构造

输入: 一个素数或素数幂 s ; 一个 $\text{OA}(n, g, p, 2)$, 记作 A ; 一个 $\text{OA}(p, f, s, 2)$, 记作 B ; 一个差阵 $D(s, s, s)$, 记作 $D = (d_0, \dots, d_{s-1})$; 一个非负整数 r ; $r+1$ 个非负整数 $g_0, g_1, \dots, g_r$, 其中 $g_0 + g_1 + \dots + g_r = g$.

输出: 一个 $\text{OD}(s^{r+1}n, \Pi_{l=0}^r(s^{r+3-l})_{g_l c_l})$, 记作 H , 其中 $c_l = \lambda s^l 2^{r-l+1} t^{r-l}$ ($l = 0, 1, \dots, r$), $\lambda = \lfloor f/2 \rfloor$, $t = \lfloor s/2 \rfloor$.

步骤 1: 将 A 的列划分为 $r+1$ 个部分, 记作 $A = (A_0, A_1, \dots, A_r)$, 其中 A_l 有 g_l 列. 对于 $l = 0, 1, \dots, r$, 在 A_l 的每一列中, 将水平 $v-1$ 替换为 B 的第 v 行 ($v = 1, \dots, p$), 并得到

$$E = (E_1, \dots, E_{g_l}),$$

其中 E_i 表示通过将 A_l 的第 i 列的水平替换为 B 的行而获得的 f 个列.

步骤 2: 对于 $i = 1, \dots, g_l$, 将每个 E_i 划分为 λ 或 $\lambda+1$ 个组, 并得到

$$E_i = (E_{i,1}, \dots, E_{i,\lambda}), \text{ 若 } f = 2\lambda \text{ 或 } E_i = (E_{i,1}, \dots, E_{i,\lambda}, t_i), \text{ 若 } f = 2\lambda + 1,$$

其中 $\lambda = \lfloor f/2 \rfloor$, 每个 $E_{i,j}$ 包含两列. 将 $E_{i,j}$ 中的所有元素视为 $GF(s)$ 中的元素. 令 $D^0(E_{i,j}) = E_{i,j}$, 并对 $1 \leq l \leq r$, 定义 $D^l(E_{i,j}) = D \oplus D^{l-1}(E_{i,j})$. 将 $g_l \lambda$ 个组

$$D^l(E_{1,1}), \dots, D^l(E_{1,\lambda}), \dots, D^l(E_{g_l,1}), \dots, D^l(E_{g_l,\lambda})$$

划分为 $g_l \lambda_l$ 个组

$$F_{1,1}, \dots, F_{1,\lambda_l}, \dots, F_{g_l,1}, \dots, F_{g_l,\lambda_l}, \text{ 其中 } \lambda_l = s^l \lambda.$$

步骤 3: 对 $l = 0, \dots, r$, 令 $k = r-l$, 并将

$$F_{1,1}, \dots, F_{1,\lambda_l}, \dots, F_{g_l,1}, \dots, F_{g_l,\lambda_l}$$

作为算法 1 步骤 2 中的输入

$$C_{1,1}, \dots, C_{1,\lambda'}, \dots, C_{g',1}, \dots, C_{g',\lambda'}.$$

根据算法 1, 可以得到一个 $\text{OD}(s^{r+1}n, (s^{r-l+3})_{g_l c_l})$, 即 $H_l = (H_{l,1}, \dots, H_{l,g_l})$, 其中 $c_l = \lambda s^l 2^{r-l+1} t^{r-l}$ ($l = 0, 1, \dots, r$), $\lambda = \lfloor f/2 \rfloor$, $t = \lfloor s/2 \rfloor$. 通过列并置将 $H_0, H_1, \dots, H_r$ 组合在一起得到

$$H = (H_0, H_1, \dots, H_r).$$

定理 4.1 由算法 2 生成的设计 H 是一个 $\text{OD}(s^{r+1}n, \Pi_{l=0}^r(s^{r+3-l})_{g_l c_l})$, 其中 $\lambda = \lfloor f/2 \rfloor$, $t = \lfloor s/2 \rfloor$, $c_l = \lambda s^l 2^{r-l+1} t^{r-l}$ ($l = 0, 1, \dots, r$). H 可以划分为 g 组, 其中前 g_0 个组每个包含 c_0 列, 接下来的 g_1 个组每个包含 c_1 列, 以此类推, 最后 g_r 个组每个包含 c_r 列, 并具有以下性质:

- (i) 任意两列实现 $s \times s^2$ 和 $s^2 \times s$ 的格子分层;
- (ii) 来自不同组的两列实现 $s \times s^3$, $s^2 \times s^2$ 和 $s^3 \times s$ 的格子分层;
- (iii) 来自两个组的三列实现 $s \times s \times s$ 的格子分层.

例 4.1 设设计 A, B, D 和 E_i ($i = 1, \dots, 5$) 与例 3.1 中的相同. 将算法 1 步骤 2 中的 $[(C_{1,1}, \dots, C_{1,\lambda'}, \dots, C_{g',1}, \dots, C_{g',\lambda'})]$ 分别取为 $[(E_1, E_2, E_3, E_4), 1]$ 和 $[(d_0 \oplus E_5, d_1 \oplus E_5), 0]$. 按照算法 1, 得到两个正交设计, 一个是 $\text{OD}(64, 16^{16})$, 记作 H_0 ; 另一个是 $\text{OD}(64, 8^4)$, 记作 H_1 . 容易验证 $H = (H_0, H_1)$ 是一个 $\text{OD}(64, 16^{16} 8^4)$, 见表 4. 它可以划分为五个组, 每组包含四列, 使得任意两列实现 2×4 和 4×2

表 4 例 4.1 中的 $OD(64, 16^{16}8^4)$, $H = (H_0, H_1)$

H_0																H_1			
-7.5	0.5	-6.5	1.5	-7.5	0.5	-6.5	1.5	-7.5	0.5	-6.5	1.5	-7.5	0.5	-6.5	1.5	-3.5	0.5	-3.5	0.5
-1.5	-6.5	-0.5	-7.5	-1.5	-6.5	-0.5	-7.5	-1.5	-6.5	-0.5	-7.5	-1.5	-6.5	-0.5	-7.5	-3.5	0.5	-3.5	0.5
1.5	6.5	0.5	7.5	1.5	6.5	0.5	7.5	1.5	6.5	0.5	7.5	1.5	6.5	0.5	7.5	-3.5	0.5	-3.5	0.5
7.5	-0.5	6.5	-1.5	7.5	-0.5	6.5	-1.5	7.5	-0.5	6.5	-1.5	7.5	-0.5	6.5	-1.5	-3.5	0.5	-3.5	0.5
-7.5	0.5	-6.5	1.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	0.5	7.5	7.5	-0.5	6.5	-1.5	-0.5	-3.5	-0.5	-3.5
-1.5	-6.5	-0.5	-7.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	1.5	6.5	0.5	7.5	-0.5	-3.5	-0.5	-3.5
1.5	6.5	0.5	7.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	-1.5	-6.5	-0.5	-7.5	-0.5	-3.5	-0.5	-3.5
7.5	-0.5	6.5	-1.5	1.5	6.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	-7.5	0.5	-6.5	1.5	-0.5	-3.5	-0.5	-3.5
-7.5	0.5	-6.5	1.5	1.5	6.5	0.5	7.5	7.5	-0.5	6.5	-1.5	-1.5	-6.5	-0.5	-7.5	0.5	3.5	0.5	3.5
-1.5	-6.5	-0.5	-7.5	7.5	-0.5	6.5	-1.5	1.5	6.5	0.5	7.5	-7.5	0.5	-6.5	1.5	0.5	3.5	0.5	3.5
1.5	6.5	0.5	7.5	-7.5	0.5	-6.5	1.5	-1.5	-6.5	-0.5	-7.5	7.5	-0.5	6.5	-1.5	0.5	3.5	0.5	3.5
7.5	-0.5	6.5	-1.5	-1.5	-6.5	-0.5	-7.5	-7.5	0.5	-6.5	1.5	1.5	6.5	0.5	7.5	0.5	3.5	0.5	3.5
-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	0.5	7.5	3.5	-0.5	3.5	-0.5
-1.5	-6.5	-0.5	-7.5	1.5	6.5	0.5	7.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	3.5	-0.5	3.5	-0.5
1.5	6.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	3.5	-0.5	3.5	-0.5
7.5	-0.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	1.5	6.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	3.5	-0.5	3.5	-0.5
-6.5	1.5	7.5	-0.5	-6.5	1.5	7.5	-0.5	-6.5	1.5	7.5	-0.5	-6.5	1.5	7.5	-0.5	-3.5	0.5	3.5	-0.5
-0.5	-7.5	1.5	6.5	-0.5	-7.5	1.5	6.5	-0.5	-7.5	1.5	6.5	-0.5	-7.5	1.5	6.5	-3.5	0.5	3.5	-0.5
0.5	7.5	-1.5	-6.5	0.5	7.5	-1.5	-6.5	0.5	7.5	-1.5	-6.5	0.5	7.5	-1.5	-6.5	-3.5	0.5	3.5	-0.5
6.5	-1.5	-7.5	0.5	6.5	-1.5	-7.5	0.5	6.5	-1.5	-7.5	0.5	6.5	-1.5	-7.5	0.5	-3.5	0.5	3.5	-0.5
-6.5	1.5	7.5	-0.5	-0.5	-7.5	1.5	6.5	0.5	7.5	-1.5	-6.5	6.5	-1.5	-7.5	0.5	-0.5	-3.5	0.5	3.5
-0.5	-7.5	1.5	6.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	0.5	7.5	-1.5	-6.5	-0.5	-3.5	0.5	3.5
0.5	7.5	-1.5	-6.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	-0.5	-7.5	1.5	6.5	-0.5	-3.5	0.5	3.5
6.5	-1.5	-7.5	0.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	-6.5	1.5	7.5	-0.5	-0.5	-3.5	0.5	3.5
-6.5	1.5	7.5	-0.5	0.5	7.5	-1.5	-6.5	6.5	-1.5	-7.5	0.5	-0.5	-7.5	1.5	6.5	0.5	3.5	-0.5	-3.5
-0.5	-7.5	1.5	6.5	6.5	-1.5	-7.5	0.5	0.5	7.5	-1.5	-6.5	-6.5	1.5	7.5	-0.5	0.5	3.5	-0.5	-3.5
0.5	7.5	-1.5	-6.5	-6.5	1.5	7.5	-0.5	-0.5	-7.5	1.5	6.5	6.5	-1.5	-7.5	0.5	0.5	3.5	-0.5	-3.5
6.5	-1.5	-7.5	0.5	-0.5	-7.5	1.5	6.5	-6.5	1.5	7.5	-0.5	0.5	7.5	-1.5	-6.5	0.5	3.5	-0.5	-3.5
-6.5	1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	-0.5	-7.5	1.5	6.5	0.5	7.5	-1.5	-6.5	3.5	-0.5	-3.5	0.5
-0.5	-7.5	1.5	6.5	0.5	7.5	-1.5	-6.5	-6.5	1.5	7.5	-0.5	6.5	-1.5	-7.5	0.5	3.5	-0.5	-3.5	0.5
0.5	7.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	3.5	-0.5	-3.5	0.5
6.5	-1.5	-7.5	0.5	-6.5	1.5	7.5	-0.5	0.5	7.5	-1.5	-6.5	-0.5	-7.5	1.5	6.5	3.5	-0.5	-3.5	0.5
-3.5	4.5	-2.5	5.5	-3.5	4.5	-2.5	5.5	-3.5	4.5	-2.5	5.5	-3.5	4.5	-2.5	5.5	-1.5	2.5	-1.5	2.5
-5.5	-2.5	-4.5	-3.5	-5.5	-2.5	-4.5	-3.5	-5.5	-2.5	-4.5	-3.5	-5.5	-2.5	-4.5	-3.5	-1.5	2.5	-1.5	2.5
5.5	2.5	4.5	3.5	5.5	2.5	4.5	3.5	5.5	2.5	4.5	3.5	5.5	2.5	4.5	3.5	-1.5	2.5	-1.5	2.5
3.5	-4.5	2.5	-5.5	3.5	-4.5	2.5	-5.5	3.5	-4.5	2.5	-5.5	3.5	-4.5	2.5	-5.5	-1.5	2.5	-1.5	2.5
-3.5	4.5	-2.5	5.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	4.5	3.5	3.5	-4.5	2.5	-5.5	-2.5	-1.5	-2.5	-1.5
-5.5	-2.5	-4.5	-3.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	5.5	2.5	4.5	3.5	-2.5	-1.5	-2.5	-1.5

(续表)

H_0										H_1									
5.5	2.5	4.5	3.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	-5.5	-2.5	-4.5	-3.5	-2.5	-1.5	-2.5	-1.5
3.5	-4.5	2.5	-5.5	5.5	2.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	-3.5	4.5	-2.5	5.5	-2.5	-1.5	-2.5	-1.5
-3.5	4.5	-2.5	5.5	5.5	2.5	4.5	3.5	3.5	-4.5	2.5	-5.5	-5.5	-2.5	-4.5	-3.5	2.5	1.5	2.5	1.5
-5.5	-2.5	-4.5	-3.5	3.5	-4.5	2.5	-5.5	5.5	2.5	4.5	3.5	-3.5	4.5	-2.5	5.5	2.5	1.5	2.5	1.5
5.5	2.5	4.5	3.5	-3.5	4.5	-2.5	5.5	-5.5	-2.5	-4.5	-3.5	3.5	-4.5	2.5	-5.5	2.5	1.5	2.5	1.5
3.5	-4.5	2.5	-5.5	-5.5	-2.5	-4.5	-3.5	-3.5	4.5	-2.5	5.5	5.5	2.5	4.5	3.5	2.5	1.5	2.5	1.5
-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	4.5	3.5	1.5	-2.5	1.5	-2.5
-5.5	-2.5	-4.5	-3.5	5.5	2.5	4.5	3.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	1.5	-2.5	1.5	-2.5
5.5	2.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	1.5	-2.5	1.5	-2.5
3.5	-4.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	5.5	2.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	1.5	-2.5	1.5	-2.5
-2.5	5.5	3.5	-4.5	-2.5	5.5	3.5	-4.5	-2.5	5.5	3.5	-4.5	-2.5	5.5	3.5	-4.5	-1.5	2.5	1.5	-2.5
-4.5	-3.5	5.5	2.5	-4.5	-3.5	5.5	2.5	-4.5	-3.5	5.5	2.5	-4.5	-3.5	5.5	2.5	-1.5	2.5	1.5	-2.5
4.5	3.5	-5.5	-2.5	4.5	3.5	-5.5	-2.5	4.5	3.5	-5.5	-2.5	4.5	3.5	-5.5	-2.5	-1.5	2.5	1.5	-2.5
2.5	-5.5	-3.5	4.5	2.5	-5.5	-3.5	4.5	2.5	-5.5	-3.5	4.5	2.5	-5.5	-3.5	4.5	-1.5	2.5	1.5	-2.5
-2.5	5.5	3.5	-4.5	-4.5	-3.5	5.5	2.5	4.5	3.5	-5.5	-2.5	2.5	-5.5	-3.5	4.5	-2.5	-1.5	2.5	1.5
-4.5	-3.5	5.5	2.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	4.5	3.5	-5.5	-2.5	-2.5	-1.5	2.5	1.5
4.5	3.5	-5.5	-2.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	-4.5	-3.5	5.5	2.5	-2.5	-1.5	2.5	1.5
2.5	-5.5	-3.5	4.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	-2.5	5.5	3.5	-4.5	-2.5	-1.5	2.5	1.5
-2.5	5.5	3.5	-4.5	4.5	3.5	-5.5	-2.5	2.5	-5.5	-3.5	4.5	-4.5	-3.5	5.5	2.5	2.5	1.5	-2.5	-1.5
-4.5	-3.5	5.5	2.5	2.5	-5.5	-3.5	4.5	4.5	3.5	-5.5	-2.5	-2.5	5.5	3.5	-4.5	2.5	1.5	-2.5	-1.5
4.5	3.5	-5.5	-2.5	-2.5	5.5	3.5	-4.5	-4.5	-3.5	5.5	2.5	2.5	-5.5	-3.5	4.5	2.5	1.5	-2.5	-1.5
2.5	-5.5	-3.5	4.5	-4.5	-3.5	5.5	2.5	-2.5	5.5	3.5	-4.5	4.5	3.5	-5.5	-2.5	2.5	1.5	-2.5	-1.5
-2.5	5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	-4.5	-3.5	5.5	2.5	4.5	3.5	-5.5	-2.5	1.5	-2.5	-1.5	2.5
-4.5	-3.5	5.5	2.5	4.5	3.5	-5.5	-2.5	-2.5	5.5	3.5	-4.5	2.5	-5.5	-3.5	4.5	1.5	-2.5	-1.5	2.5
4.5	3.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	1.5	-2.5	-1.5	2.5
2.5	-5.5	-3.5	4.5	-2.5	5.5	3.5	-4.5	4.5	3.5	-5.5	-2.5	-4.5	-3.5	5.5	2.5	1.5	-2.5	-1.5	2.5

的格子分层, 来自不同组的两列实现 2×8 , 4×4 和 8×2 的格子分层, 来自两个组的三列实现 $2 \times 2 \times 2$ 的格子分层.

表 5 总结了算法 2 生成的混合水平正交设计. 这些设计的空间填充性由参数 g (组数), $g_0, \dots, g_r$ (具有相同水平数的因子的组数), $c_0, \dots, c_r$ (每组中的列数) 和 π (实现 $s \times s^3$, $s^2 \times s^2$ 和 $s^3 \times s$ 格子分层的二维子矩阵的比例) 来表征.

5 应用与比较

我们通过一个应用来说明所得设计的有用性. 具体而言, 我们评估了所得设计在构建统计替代模型方面的表现, 并与其他类型的设计进行了比较. 我们运行计算机模拟代码, 并根据钻孔函数

$$f(X) = \frac{2\pi x_3(x_4 - x_6)}{\ln(x_2 - x_1)[1 + 2x_3x_7/(\ln(x_2/x_1)x_1^2x_8) + x_3/x_5]}$$

表 5 算法 2 生成的混合水平空间填充正交设计

$A: OA(n, g, p, 2)$	$B: OA(p, f, s, 2)$	r	$(g_0, \dots, g_r)$	$W: OD(s^{r+1}n, \prod_{l=0}^r (s^{r+3-l})^{g_l c_l})$	$(c_0, \dots, c_r)$	π (%)
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	1	(1,4)	OD(64, 16 ⁴ 8 ¹⁶) [†]	(4,4)	84.21
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	1	(2,3)	OD(64, 16 ⁸ 8 ¹²) [†]	(4,4)	84.21
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	1	(3,2)	OD(64, 16 ¹² 8 ⁸) [†]	(4,4)	84.21
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	1	(4,1)	OD(64, 16 ¹⁶ 8 ⁴) [†]	(4,4)	84.21
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(1,0,4)	OD(128, 32 ⁸ 8 ³²) [†]	(8,0,8)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(1,1,3)	OD(128, 32 ⁸ 16 ⁸ 8 ²⁴) [†]	(8,8,8)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(1,2,2)	OD(128, 32 ⁸ 16 ¹⁶ 8 ¹⁶) [†]	(8,8,8)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(1,3,1)	OD(128, 32 ⁸ 16 ²⁴ 8 ⁸) [†]	(8,8,8)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(1,4,0)	OD(128, 32 ⁸ 16 ³²) [†]	(8,8,0)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(2,0,3)	OD(128, 32 ¹⁶ 8 ²⁴) [†]	(8,0,8)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(2,1,2)	OD(128, 32 ¹⁶ 16 ⁸ 8 ¹⁶) [†]	(8,8,8)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(2,3,0)	OD(128, 32 ¹⁶ 16 ²⁴) [†]	(8,8,0)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(3,1,1)	OD(128, 32 ²⁴ 16 ⁸ 8 ⁸) [†]	(8,8,8)	82.05
OA(16, 5, 4, 2)	OA(4, 3, 2, 2)	2	(4,1,0)	OD(128, 32 ³² 16 ⁸) [†]	(8,8,0)	82.05
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(1,9)	OD(729, 81 ⁸ 27 ¹⁰⁸)	(8,12)	90.67
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(2,8)	OD(729, 81 ¹⁶ 27 ⁹⁶)	(8,12)	90.60
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(3,7)	OD(729, 81 ²⁴ 27 ⁸⁴)	(8,12)	90.55
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(4,6)	OD(729, 81 ³² 27 ⁷²)	(8,12)	90.52
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(5,5)	OD(729, 81 ⁴⁰ 27 ⁶⁰)	(8,12)	90.51
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(6,4)	OD(729, 81 ⁴⁸ 27 ⁴⁸)	(8,12)	90.53
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(7,3)	OD(729, 81 ⁵⁶ 27 ³⁶)	(8,12)	90.59
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(8,2)	OD(729, 81 ⁶⁴ 27 ²⁴)	(8,12)	90.70
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	1	(9,1)	OD(729, 81 ⁷² 27 ¹²)	(8,12)	90.88
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(1,0,9)	OD(2187, 243 ¹⁶ 27 ³²⁴)	(16,0,36)	89.95
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(1,1,8)	OD(2187, 243 ¹⁶ 81 ²⁴ 27 ²⁸⁸)	(16,24,36)	89.86
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(1,5,4)	OD(2187, 243 ¹⁶ 81 ¹²⁰ 27 ¹⁴⁴)	(16,24,36)	89.71
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(1,4,5)	OD(2187, 243 ¹⁶ 81 ⁹⁶ 27 ¹⁸⁰)	(16,24,36)	89.70
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(1,9,0)	OD(2187, 243 ¹⁶ 81 ²¹⁶)	(16,24,0)	90.28
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(2,0,8)	OD(2187, 243 ³² 27 ²⁸⁸)	(16,24,36)	89.66
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(2,4,4)	OD(2187, 243 ³² 81 ⁹⁶ 27 ¹⁴⁴)	(16,24,36)	89.52
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(2,8,0)	OD(2187, 243 ³² 81 ¹⁹²)	(16,24,0)	90.20
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(3,3,4)	OD(2187, 243 ⁴⁸ 81 ⁷² 27 ¹⁴⁴)	(16,24,36)	89.32
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(4,3,3)	OD(2187, 243 ⁶⁴ 81 ⁷² 27 ¹⁰⁸)	(16,24,36)	89.21
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(5,2,3)	OD(2187, 243 ⁸⁰ 81 ⁴⁸ 27 ¹⁰⁸)	(16,24,36)	89.03
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(6,2,2)	OD(2187, 243 ⁹⁶ 81 ⁴⁸ 27 ⁷²)	(16,24,36)	89.10
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(7,1,2)	OD(2187, 243 ¹¹² 81 ²⁴ 27 ⁷²)	(16,24,36)	88.96
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(8,1,1)	OD(2187, 243 ¹²⁸ 81 ²⁴ 27 ³⁶)	(16,24,36)	89.38
OA(81, 10, 9, 2)	OA(9, 4, 3, 2)	2	(9,0,1)	OD(2187, 243 ¹⁴⁴ 27 ³⁶)	(16,0,36)	89.39

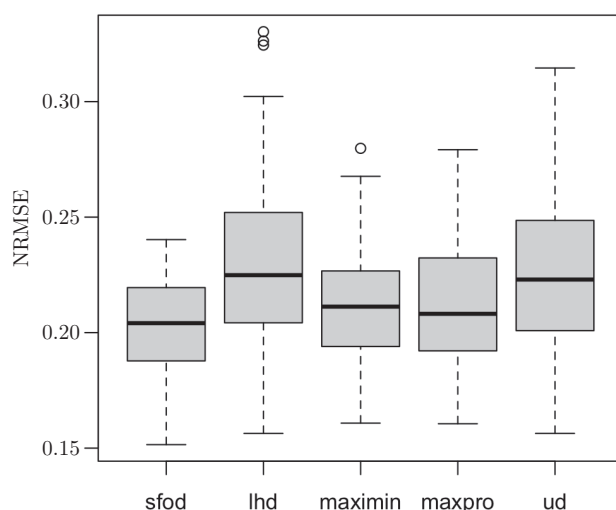


图 3 空间填充正交设计 $OD(32, 8^8)$ (sfod)、随机拉丁超立方体设计 (lhd)、最大最小拉丁超立方体的设计 (maximin)、最大投影拉丁超立方体设计 (maxpro) 和均匀设计 (ud) 的规范化均方根误差 (NRMSE)

生成训练数据, 其中 $X = (x_1, \dots, x_8)$ 为输入变量. 钻孔函数广泛应用于评估计算机试验设计的模拟中 (参见文献 [16, 21, 31]). 我们通过拟合 Gauss 过程模型来近似钻孔函数, 采用常数均值和参数 $\nu = 5/2$ 的 Matern 相关函数 (参见文献 [33]). 然后通过规范化均方根误差

$$\text{NRMSE} = \left[\frac{\sum_{X_i \in \psi} (\hat{y}(X_i) - y(X_i))^2}{\sum_{X_i \in \psi} (\bar{y} - y(X_i))^2} \right]^{1/2}$$

来评估不同类型设计的表现, 其中 $\hat{y}(X_i)$ 是拟合 Gauss 过程模型的预测响应, $\bar{y}$ 是用于构建模型的训练数据的平均响应, ψ 是测试数据集.

我们使用算法 1 构造的空间填充正交设计 $OD(32, 8^8)$ (例 3.1 中设计 $OD(32, 8^{10})$ 的前 8 列) 来构建统计替代模型, 并将其与如下四种其他类型的设计进行比较:

- (i) 来自 R 包 lhs 的随机拉丁超立方体设计 (参见文献 [3]);
- (ii) 来自 R 包 SLHD 的最大最小拉丁超立方体设计 (参见文献 [1]);
- (iii) 来自 R 包 MaxPro 的最大投影拉丁超立方体设计 (参见文献 [11]);
- (iv) 来自 R 包 UniDOE 的均匀设计 (参见文献 [37]).

这些设计都包含 32 次试验和 8 个因子. 对于每个设计, 我们将因子的水平缩放到 $[0, 1]$ 上, 并进一步考虑因子的列置换. 图 3 显示了 100 个随机列置换对应的规范化均方根误差的箱线图, 其中测试数据集是一个试验次数为 1,000 的随机拉丁超立方体设计. 显然, 本文构造的空间填充正交设计优于所有其他设计.

6 结论

空间填充性和正交性是计算机试验设计的两个重要性质. 空间填充性适用于因子筛选, 而正交性可以保证主效应的估计互不相关. 文献中已经提出了许多构造空间填充设计和正交设计的方法, 而对于同时具有这两种性质的设计的研究却很少. 本文提出了构造空间填充正交设计的一般方法. 该方法

基于半旋转矩阵且易于实现. 相关理论支持已得到充分论证. 此外, 所得设计在试验次数和水平数上具有灵活性, 使得这类设计非常适用于计算机试验.

致谢 作者衷心感谢审稿人提出的建设性意见.

参考文献

- 1 Ba S, Myers W R, Brennenman W A. Optimal sliced Latin hypercube designs. *Technometrics*, 2015, 57: 479–487
- 2 Bingham D, Sitter R R, Tang B. Orthogonal and nearly orthogonal designs for computer experiments. *Biometrika*, 2009, 96: 51–65
- 3 Carnell R. lhs: Latin hypercube samples. R package version 1.1.5. <https://cran.r-project.org/web/packages/lhs/index.html>, 2022
- 4 Fang K T, Li R, Sudjianto A. *Design and Modeling for Computer Experiments*. New York: Chapman and Hall/CRC, 2006
- 5 Georgiou S D, Stylianou S, Drosou K, et al. Construction of orthogonal and nearly orthogonal designs for computer experiments. *Biometrika*, 2014, 101: 741–747
- 6 He X, Qian P Z. A central limit theorem for nested or sliced Latin hypercube designs. *Statist Sinica*, 2016, 26: 1117–1128
- 7 He Y, Cheng C S, Tang B. Strong orthogonal arrays of strength two plus. *Ann Statist*, 2018, 46: 457–468
- 8 He Y, Tang B. Strong orthogonal arrays and associated Latin hypercubes for computer experiments. *Biometrika*, 2013, 100: 254–260
- 9 He Y, Tang B. A characterization of strong orthogonal arrays of strength three. *Ann Statist*, 2014, 42: 1347–1360
- 10 Hedayat A S, Sloane N J A, Stufken J. *Orthogonal Arrays: Theory and Applications*. New York: Springer, 1999
- 11 Joseph V R, Gul E, Ba S. Maximum projection designs for computer experiments. *Biometrika*, 2015, 102: 371–380
- 12 Li W L, Liu M Q, Yang J F. Several new classes of space-filling designs. *Statist Papers*, 2024, 65: 357–379
- 13 Lin C D, Mukerjee R, Tang B. Construction of orthogonal and nearly orthogonal Latin hypercubes. *Biometrika*, 2009, 96: 243–247
- 14 Liu H, Liu M Q. Column-orthogonal strong orthogonal arrays and sliced strong orthogonal arrays. *Statist Sinica*, 2015, 25: 1713–1734
- 15 McKay M D, Beckman R J, Conover W J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 1979, 21: 239–245
- 16 Moon H, Dean A M, Santner T J. Two-stage sensitivity-based group screening in computer experiments. *Technometrics*, 2012, 54: 376–387
- 17 Mukerjee R, Sun F, Tang B. Nearly orthogonal arrays mappable into fully orthogonal arrays. *Biometrika*, 2014, 101: 957–963
- 18 Owen A B. Orthogonal arrays for computer experiments, integration and visualization. *Statist Sinica*, 1992, 2: 439–452
- 19 Santner T J, Williams B J, Notz W I. *The Design and Analysis of Computer Experiments*, 2nd ed. New York: Springer, 2018
- 20 Shi C, Tang B. Construction results for strong orthogonal arrays of strength three. *Bernoulli*, 2020, 26: 418–431
- 21 Shi C, Xu H. A projection space-filling criterion and related optimality results. *J Amer Statist Assoc*, 2024, 119: 2658–2669
- 22 Steinberg D M, Lin D K J. A construction method for orthogonal Latin hypercube designs. *Biometrika*, 2006, 93: 279–288
- 23 Sun F S, Liu M Q, Lin D K J. Construction of orthogonal Latin hypercube designs. *Biometrika*, 2009, 96: 971–974
- 24 Sun F S, Tang B. A general rotation method for orthogonal Latin hypercubes. *Biometrika*, 2017, 104: 465–472
- 25 Sun F S, Tang B. A method of constructing space-filling orthogonal designs. *J Amer Statist Assoc*, 2017, 112: 683–689
- 26 Tang B. Orthogonal array-based Latin hypercubes. *J Amer Statist Assoc*, 1993, 88: 1392–1397
- 27 Tian Y, Xu H. A minimum aberration-type criterion for selecting space-filling designs. *Biometrika*, 2021, 109: 489–501
- 28 Wang C, Lin D K J, Liu M Q. A new class of orthogonal designs with good low dimensional space-filling properties. *Statist Sinica*, 2025, 35
- 29 Wang C, Yang J, Liu M Q. Construction of space-filling orthogonal designs. *J Statist Plann Inference*, 2021, 213: 130–141
- 30 Wang C, Yang J, Liu M Q. Construction of strong group-orthogonal arrays. *Statist Sinica*, 2022, 32: 1225–1243
- 31 Wang Y, Sun F. On the maximin distance properties of orthogonal designs via the rotation. *Sci China Math*, 2023, 66: 1593–1608

- 32 Wu C F J, Hamada M. Experiments: Planning, Analysis, and Optimization, 3rd ed. New York: Wiley, 2021
- 33 Xiao Q, Wang L, Xu H. Application of kriging models for a drug combination experiment on lung cancer. *Stat Med*, 2019, 38: 236–246
- 34 Xie K H, Xiong S, Qian P Z, et al. General sliced Latin hypercube designs. *Statist Sinica*, 2014, 24: 1239–1256
- 35 Yang J, Liu M Q. Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs. *Statist Sinica*, 2012, 22: 433–442
- 36 Ye K Q. Orthogonal column Latin hypercubes and their application in computer experiments. *J Amer Statist Assoc*, 1998, 93: 1430–1439
- 37 Zhang A, Li H, Quan S, et al. UniDOE: Uniform design of experiments. R package version 1.0.2. <https://cran.r-project.org/package=UniDOE>, 2018
- 38 Zhou Y, Tang B. Column-orthogonal strong orthogonal arrays of strength two plus and three minus. *Biometrika*, 2019, 106: 997–1004

附录 A

附录 A.1 定理 3.1 的证明

首先证明 W 是一个 $\text{OD}(s^{k+1}n, (s^{k+3})^{gc})$. 对 $i = 1, 2, \dots, g, j = 1, 2, \dots, \lambda, r_1, \dots, r_k = 0, 1, \dots, t-1$, 每个 $\tilde{E}_{i,j}^{r_1 r_2 \dots r_k}$ 有 $s^{k+1}n$ 行 2^{k+2} 列, 并可以通过 R_k 旋转为一个大小为 $s^{k+1}n \times 2^{k+1}$ 的正交设计 $\tilde{E}_{i,j}^{r_1 r_2 \dots r_k} R_k^T$. 因此, 来自同一个 $\tilde{E}_{i,j}^{r_1 r_2 \dots r_k} R_k^T$ 的两列是正交的.

$$(\tilde{E}_{1,1}^{00\dots 0}, \dots, \tilde{E}_{1,1}^{(t-1)(t-1)\dots(t-1)}, \dots, \tilde{E}_{g,\lambda}^{00\dots 0}, \dots, \tilde{E}_{g,\lambda}^{(t-1)(t-1)\dots(t-1)})$$

是一个强度为 2 的 s 水平正交表. 因此, 来自不同 $\tilde{E}_{i,j}^{r_1 r_2 \dots r_k} R_k^T$ 的两列也是正交的. 则设计 W 是一个

$$\text{OD}(s^{k+1}n, (s^{k+3})^{gc}).$$

接下来, 考虑设计 W 的空间填充性. 根据文献 [28, 定理 3], 可知定理 3.1 在 $k=0$ 时成立, 即当 $k=0$ 时, 基于正交表 A, B 和 k 的设计 W 是一个 $\text{OD}(sn, (s^3)^{gc})$, 其中

$$c = 2\lambda, \quad \lambda = \lfloor f/2 \rfloor.$$

W 可以划分为 g 组, 每组包含 c 列, 任意两列实现 $s \times s^2$ 和 $s^2 \times s$ 的格子分层, 来自不同组的两列实现 $s \times s^3, s^2 \times s^2$ 和 $s^3 \times s$ 的格子分层, 来自两个组的三列实现 $s \times s \times s$ 的格子分层. 下面考虑 $k \geq 1$ 的情形. 对 $i = 1, 2, \dots, g, j = 1, 2, \dots, \lambda, r_1, \dots, r_k = 0, 1, \dots, t-1$, 当 $k \geq 1$ 时, 设计 $E_{i,j}^{r_1 r_2 \dots r_k}$ 由 $C_{i,j}$ 通过 D 进行扩展得到. 因此, 基于 $k=0$ 构造的正交设计的二维和三维空间填充性也适用于基于 $k \geq 1$ 构造的正交设计.

附录 A.2 定理 3.2 的证明

给定 $\lfloor f/2 \rfloor = 1$ 和 $\lfloor s/2 \rfloor = 1$, 算法 1 构造的设计 W 是一个 $\text{OD}(s^{k+1}n, (s^{k+3})^{gc})$, 其中 $c = 2^{k+1}$. 设 x 和 y 是来自不同类的两列, 将 x 和 y 的 s^{k+3} 个水平分别塌陷为 s 和 s^3 水平后, (x, y) 变为 $(x_1, s^2 y_1 + s y_2 + y_3)$. 可以验证, 在某个行置换下,

$$\{x_1, y_1, y_2, y_3\} = \{d_0 \oplus z_1, d_0 \oplus z_2, d_0 \oplus z_3, d_1 \oplus z_3\},$$

其中 (z_1, z_2, z_3) 是一个 $\text{OA}(s^k n, 3, s, 3)$, d_0 和 d_1 分别表示 D 的第一和二列. 根据文献 [28, 引理 1], 可知 (x_1, y_1, y_2, y_3) 是一个 $\text{OA}(s^{k+1}n, 4, s, 4)$. 因此, 对设计 W , 来自不同类的任意两列实现 $s \times s^3$ 和 $s^3 \times s$ 的格子分层.

New results for space-filling orthogonal designs

Chunyan Wang & Fasheng Sun

Abstract Computer experiments are popularly used to study complicated systems in many fields. Orthogonality and space-filling properties are two important design properties for such experiments. Space-filling properties guarantee effective factor screening while the orthogonality allows the estimates of the main effects in linear models to be uncorrelated with each other. In this paper, we develop a general rotation method for constructing a new class of orthogonal designs. The proposed method is easy to operate and flexible in both run size and number of levels. The resulting designs are not only orthogonal but also entertain attractive space-filling properties. These designs can be either symmetric or asymmetric. Related theoretical supports are well established. The numerical results validate the advantages of the proposed designs.

Keywords computer experiment, group space-filling design, orthogonality, semi-rotation matrix

MSC(2020) 62K15, 62K99

doi: 10.1360/SCM-2024-0514