应用概率统计 第 38 卷
第 2 期 2022 年 4 月

Chinese Journal of Applied Probability and Statistics
Apr., 2022, Vol. 38, No. 2, pp. 253-266
doi: 10.3969/j.issn.1001-4268.2022.02.006

复杂约束域内混料最优设计的格点评价 *

李光辉 (凯里学院理学院, 凯里, 556011)

李俊鹏 张崇岐* (广州大学经济与统计学院, 广州, 510006)

摘 要: 在具有复杂约束的混料试验域内, 要证明一个设计的最优性是困难的, 但要否定设计的最优性却相对容易. 本文通过构造复杂约束域内的稠密格点, 不仅可以用于评价一个设计的最优性, 还可以比较多个局部最优设计的优劣, 并且稠密覆盖的格子点集作为备选点集可以产生约束区域上的局部最优设计. 通过实例分析, 这种方法是有效的.

关键词: 混料试验; 最优设计; 格子点集

中图分类号: O212.6

英文引用格式: LI G H, LI J P, ZHANG C Q. Lattice evaluation of the optimal mixture design in the complex constraint region [J]. Chinese J Appl Probab Statist, 2022, 38(2): 253–266. (in Chinese)

§1. 引 言

随着科学技术的发展, 在生产实践中经常需要进行大型试验来获取数据. 大型试验是指多因子、多指标、多约束, 并且因子之间存在各种复杂的交互作用的试验. 有的大型混料试验由于操作复杂, 周期漫长, 并且一次试验就会耗费大量的人力和财力. 为了节省成本, 试验者往往先通过计算机模拟试验来筛选因子和试验点, 构造稳健的模型, 然后再应用于实际中. 在很多混料试验中, 各分量除了要满足基本约束条件, 往往还会受到各种复杂的约束的限制, 这就使得试验的可行域是单纯形内部的不规则几何体, 这个几何体的边界面有可能是线性的, 也有可能是非线性的, 有的区域可能是凸的, 也可能是非凸的. 因此, 求解复杂约束试验域内的混料最优设计是比较困难的, 这一问题也是该领域研究的热点. 混料最优设计是以凸设计理论为基础, 在给定的模型下, 将信息阵的凸函数进行优化的一类设计. 关于凸试验域的最优设计问题主要有两类方法.

一类是根据 Elfving [1] 的思想, Mclean 和 Anderson [2] 提出一种渐近最优的极端顶点设计. 1974 年, Snee 和 Marquardt [3] 提出了 XVERT 算法用于搜索出具有上下界约束的混料试验域的极端顶点. 针对线性约束混料试验域, Snee [4, 5] 系统的研究了极端顶点搜索问

*国家自然科学基金项目 (批准号: 11901260、12071096)、贵州省科学技术基金项目 (批准号: 黔科合基础 [2020]1Y010) 和凯里学院博士专项课题 (批准号: BS201807) 资助.

*通讯作者, E-mail: cqzhang@gzhu.edu.cn.

本文 2020 年 9 月 30 日收到, 2020 年 12 月 9 日收到修改稿.

题, 提出了 CONSIM 算法. 关颖男[6] 在此基础上对 CONSIM 算法进行了改进, 提出了针对约束混料区域搜索极端顶点的分枝–限定算法, 此方法使用简单, 而且规格化, 适于计算机计算, 也能避免 XVERT 算法可能产生重复顶点这样的缺点.

另一类方法是使用优化算法来搜索最优设计. 近年来, 一些统计学者研究使用各类优化算法来搜索各类混料模型在不同约束域内的最优设计, Karunanithi 等[7] 使用 CAMD 算法搜索混料试验的 D-最优设计. Limmun 等[8] 使用遗传算法搜索出了多项式模型的 D-和 Ds-最优设计. Wong 等[9] 利用 PSO 算法得到了一些三分量常用的模型的 D-, A-最优设计. Li 和 Zhang[10] 构造了一类随机搜索算法, 用于搜索复杂约束域上的渐近最优设计. 目前, 以上研究成果都已经应用于实践中, 在一些常见统计软件中也配套了相应的程序包.

不论是使用凸区域的线性规划方式计算最优设计, 还是使用优化算法搜索最优设计, 当试验域所受的约束过于复杂时, 求解全局最优设计都是困难的. 为此, 有以下几个方面是值得研究的:

1. 使用算法搜索出的设计极易落入局部最优, 许多学者也提出了一系列改进措施力求搜索出全局最优解, 但当混料约束区域十分复杂时仍无法避免. 局部最优解与初值的设定以及迭代过程有关, 产生的结果也可能有许多个. 虽然我们难以证明一个设计在复杂约束域上是全局最优的, 但可以通过找到不满足最优性等价条件[11] 的设计点来否定一个设计的最优性. 对于若干个局部最优设计, 如何从中挑选出 “最好” 的设计是值得研究的.
2. 很多情形下, 试验者并不明确真实的模型是为何, 或者不确定是几类模型中的哪一种, 直接使用均匀设计又会损失边界的诸多信息. 因此, 如何选择一个合适的设计能够兼顾几类模型的最优性也是十分重要的.
3. 特别是有的试验域受非线性约束的限制, 形成复杂的区域乃至非凸试验域. 这就使得试验域不存在极端顶点, 在边界搜索又难以实现, 这时就需要在约束区域构造高密度的备选点集, 并从中选择出近似最优的结果.

使用优化算法可以搜索出复杂约束域上的局部最优设计, 如何评价这些设计的优劣, 除了比较它们在各准则下的最优性表现, 还应从全局上比较它们的预测方差, 从而对各设计进行整体比较. 本文以混料格子点集为基础, 构造复杂约束的试验域上的稠密格子点覆盖, 由此构造的格子点集 NT-net 在度量设计的均匀性, 评价模型的预测效果以及比较多个局部最优设计的优劣方面都有很好的应用.

本文首先介绍混料模型与最优准则, 第 3 节基于格子点集提出了复杂约束域上的稠密格子点集覆盖的原理与方法; 第 4 节讨论了使用格子点集来检测一个设计是否为全局最优的方法, 并用它来评价多个局部最优设计的优劣; 第 5 节讨论了基于稠密覆盖的格子点集生成局部最优设计的方法; 第 6 节给出了实例分析, 由此可见这种方法是易行有效的; 最后提出了进一步可研究的问题.

§2. 模型与最优准则

对于 q 分量混料系统, 响应是各分量比例 $x_1, x_2, \cdots, x_q$ 的函数, 由各分量比例所确定的试验域 $\mathscr{X}$ 可表示为

$$\mathscr{X} = \Big\{\boldsymbol{x} = (x_1, x_2, \cdots, x_q) : \sum_{i=1}^{q} x_i = 1,\ x_i \geqslant 0,\ i = 1, 2, \cdots, q,\ C's\Big\},$$

其中 $C's : a_j \leqslant \varphi_j(x_1, x_2, \cdots, x_q) \leqslant b_j$ 表示附加约束条件, $\varphi_j(x_1, x_2, \cdots, x_q)$ 是关于各分量的函数, a_j, b_j, $j = 1, 2, \cdots, k$ 是已知的常数. 如果试验域 $\mathscr{X}$ 不含附加约束条件 $C's$, 则 $\mathscr{X} = S^{q-1}$ 是一个 $q-1$ 维的正规单纯形.

最优设计一般都是建立在模型已知的情形下, 假设我们考虑的一般线性模型的基本形式为

$$y = \beta^{\mathsf{T}} \boldsymbol{f}(\boldsymbol{x}) + \varepsilon, \tag{1}$$

其中 y 是响应变量, $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))^{\mathsf{T}}$ 是给定的关于 $\boldsymbol{x} = (x_1, x_2, \cdots, x_q)^{\mathsf{T}} \in \mathscr{X} \subseteq S^{q-1}$ 的函数向量, $\beta = (\beta_1, \beta_2, \cdots, \beta_m)^{\mathsf{T}}$ 是 m 维的未知参数向量, ε 是随机误差, 通常假设 $\mathsf{E}(\varepsilon) = 0$, $\mathsf{Var}\,(\varepsilon) = \sigma^2$.

在模型 (1) 式的基础上定义的一个 n 点设计为

$$\xi = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n \\ w_1 & w_2 & \cdots & w_n \end{pmatrix},$$

其中 $\boldsymbol{x}_i \in \mathscr{X} \subseteq S^{q-1}$, $i = 1, 2, \cdots, n$, $\sum\limits_{i=1}^{n} w_i = 1$. 所有设计构成的全集 Ξ 称为一个设计空间, 定义设计 $\xi \in \Xi$ 的信息矩阵为

$$M(\xi) = \int_{\mathscr{X}} \boldsymbol{f}(\boldsymbol{x}) \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{x}) \xi(\mathrm{d}\boldsymbol{x}). \tag{2}$$

最优设计就是确定一个设计 $\xi^* \in \Xi$, 使得在某种意义下达到最优. 常见的最优设计都与模型的信息矩阵有关, 并且与之对应的等价性定理也是建立在关于信息矩阵的判决函数的基础上.

- 称 $\xi^* = \arg\max\limits_{\xi \in \Xi} \det(M(\xi))$ 是 D-最优设计, 并且 ξ^* 是 D-最优设计的充分必要条件:

$$\psi_D(\boldsymbol{x}, \xi^*) = \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{x}) M^{-1}(\xi^*) \boldsymbol{f}(\boldsymbol{x}) \leqslant m. \tag{3}$$

- 称 $\xi^* = \arg\min\limits_{\xi \in \Xi} \mathrm{tr}(M^{-1}(\xi))$ 是 A-最优设计, 并且 ξ^* 是 A-最优设计的充分必要条件:

$$\psi_A(\boldsymbol{x}, \xi^*) = \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{x}) M^{-2}(\xi^*) \boldsymbol{f}(\boldsymbol{x}) \leqslant \mathrm{tr}(M^{-1}(\xi^*)). \tag{4}$$

- 称 $\xi^*=\arg\min\limits_{\xi\in\Xi}\prod\limits_{i=1}^{m}(M^{-1}(\xi))_{ii}$ 是 R-最优设计, 并且 ξ^* 是 R-最优设计的充分必要条件:

$$\psi_R(\boldsymbol{x},\xi^*)=\boldsymbol{f}^{\tau}(\boldsymbol{x})M^{-1}(\xi^*)BM^{-1}(\xi^*)\boldsymbol{f}(\boldsymbol{x})\leqslant m, \tag{5}$$

其中 $B=\mathrm{diag}(1/M_{11},1/M_{22},\cdots,1/M_{mm})$, M_{ii} 是 $M^{-1}(\xi)$ 中主对角线上的第 i 个元素.

以上最优准则都是基于信息矩阵而定义的, 只有当 $\boldsymbol{x}\in\mathscr{X}$ 为柱点时不等式 (3)、(4)、(5) 式取等号. 我们称 (3)、(4)、(5) 式中的函数 $\psi_\phi(\boldsymbol{x},\xi)$ 为 ϕ-准则下的方差函数, 这里下标 ϕ 代表 D, A 及 R. 由此可见, 要判断一个设计 ξ 能否达到最优, 需要证明方差函数 $\psi_\phi(\boldsymbol{x},\xi)\leqslant 0$ 是否成立, 在约束及模型都比较复杂的情形下, 这是困难的.

§3. 稠密格点覆盖

为了讨论试验域上的最优性评价问题, 需要构造出均匀分布在约束试验域 $\mathscr{X}$ 上的格子点集, 我们将一般的格子点集的定义推广至约束试验域情形. 首先确定一个上下界约束区域与格点覆盖.

定义 1 令

$$S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}=\{\boldsymbol{x}=(x_1,x_2,\cdots,x_q):0\leqslant a_i\leqslant x_i\leqslant b_i\leqslant 1,\ i=1,2,\cdots,q,\ \boldsymbol{x}\in S^{q-1}\}$$

表示上下界约束域, 其中 $\boldsymbol{a}=(a_1,a_2,\cdots,a_q)^{\tau}$ 与 $\boldsymbol{b}=(b_1,b_2,\cdots,b_q)^{\tau}$ 称为下界与上界约束向量. 若 $\mathscr{X}\subseteq S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$, 并且对于任意的 $\varepsilon>0$ 和 $i,j=1,2,\cdots,q$, 取向量 $\boldsymbol{a'}=(a_1,\cdots,a_i+\varepsilon,\cdots,a_q)^{\tau}$ 及 $\boldsymbol{b'}=(b_1,\cdots,b_j-\varepsilon,\cdots,b_q)^{\tau}$, 都有 $\mathscr{X}\not\subset S^{q-1}_{[\boldsymbol{a'},\boldsymbol{b}]}$ 及 $\mathscr{X}\not\subset S^{q-1}_{[\boldsymbol{a},\boldsymbol{b'}]}$. 我们称 $S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$ 是 $\mathscr{X}$ 的一个最小上下界约束覆盖域, 简称为上下界覆盖域.

如果试验域的体积很小, 这意味着 $p=\mathrm{Vol}(\mathscr{X})/\mathrm{Vol}(S^{q-1})$ 越接近 0, 这里 $\mathrm{Vol}(\mathscr{X})$ 表示 $\mathscr{X}$ 的体积. 如果 p 不是很小的数, 以往生成约束试验域 $\mathscr{X}\subseteq S^{q-1}$ 内格子点集的方法是在 S^{q-1} 上产生足够高阶的格子点集, 然后再保留下落入 $\mathscr{X}$ 内的试验点即可. 如果 p 十分接近于 0, 或者约束条件 $C's$ 十分复杂, 难以求得 $\mathscr{X}$ 的上下界覆盖. 若直接使用高阶格子点集会造成很大的浪费, 特别对于体积很小的狭长形区域, 落入 $\mathscr{X}$ 中的格点是很少的.

以下分为两个阶段来实现 $\mathscr{X}$ 上的稠密格点覆盖.

第一阶段, 求出 $\mathscr{X}$ 的近似上下界覆盖. 使用高阶格子点集

$$\mathscr{L}\{q,m\}=\Big\{\Big(\frac{\alpha_1}{m},\frac{\alpha_2}{m},\cdots,\frac{\alpha_q}{m}\Big):\alpha_i\in\mathbb{Z}^+,\ i=1,2,\cdots,q,\ \sum_{i=1}^{q}\alpha_i=m\Big\}. \tag{6}$$

若 $\mathscr{L}_{\mathscr{X}}=\mathscr{L}\{q,m\}\cap\mathscr{X}\neq\emptyset$, 以 L_X 表示 $\mathscr{L}_{\mathscr{X}}$ 中的各点按行排列而成的矩阵. 当 q 较大时, 也可考虑取使用 l 个互质的整数 $n_1,n_2,\cdots,n_l$, 用 $\bigcup\limits_{j=1}^{l}\mathscr{L}\{q,n_j\}$ 来代替高阶格子点

集. 因为 $\bigcap\limits_{j=1}^{l}\mathscr{L}\{q,n_j\}=\mathscr{L}\{q,1\}$, 这样不至于产生过多重复的点. 即令

$$\mathscr{L}_{\mathscr{X}}=\bigcup_{j=1}^{l}\mathscr{L}\{q,n_j\}\cap\mathscr{X},$$

设若 $\mathscr{L}_{\mathscr{X}}$ 中元素个数为 $\#(\mathscr{L}_{\mathscr{X}})=n_0$, 令

$$\boldsymbol{x}_0=(x_{01},x_{02},\cdots,x_{0q})^{\mathsf{T}}=\frac{1}{n_0}L_X^{\mathsf{T}}\mathbf{1}_{n_0}$$

为 $\mathscr{X}$ 的近似重心.

为了确定包含 $\mathscr{X}$ 的上下界覆盖域, 采用格点划分的方法实现. 令矩阵

$$G_i(\lambda;m)=[\boldsymbol{g}_1,\cdots,\boldsymbol{g}_{i-1},(x_{0i}+\lambda)\mathbf{1}_n,\boldsymbol{g}_{i+1},\cdots,\boldsymbol{g}_q],\qquad i=1,2,\cdots,q,$$

其中 $n=\binom{q+m-2}{m}$ 为该矩阵的行数, $[\boldsymbol{g}_1,\cdots,\boldsymbol{g}_{i-1},\boldsymbol{g}_{i+1},\cdots,\boldsymbol{g}_q]=(1-x_{0i}-\lambda)L(q-1,m)$, $\mathbf{1}_n$ 为元素全为 1 的 n 维列向量. 对于任意的常数 $\varepsilon>0$, 若存在 $\underline{\lambda}_i^*\leqslant 0$ 和 $\overline{\lambda}_i^*\geqslant 0$, 使得满足

$$\begin{aligned}&G_i(\overline{\lambda}_i^*;m)\cap\mathscr{X}\neq\emptyset,\quad \text{而 } G_i(\overline{\lambda}_i^*+\varepsilon;m)\cap\mathscr{X}=\emptyset;\\&G_i(\underline{\lambda}_i^*;m)\cap\mathscr{X}\neq\emptyset,\quad \text{而 } G_i(\underline{\lambda}_i^*-\varepsilon;m)\cap\mathscr{X}=\emptyset.\end{aligned}$$

则分量 x_i 的上、下界近似为 $x_{0i}-\underline{\lambda}_i^*\leqslant x_i\leqslant x_{0i}+\overline{\lambda}_i^*$, $i=1,2,\cdots,q$. 若给定一个松弛变量 $\varepsilon_0>0$, 令

$$\begin{aligned}\boldsymbol{a}&=(a_1,a_2,\cdots,a_q)^{\mathsf{T}}=\boldsymbol{x}_0+(\underline{\lambda}_1^*-\varepsilon_0,\underline{\lambda}_2^*-\varepsilon_0,\cdots,\underline{\lambda}_q^*-\varepsilon_0)^{\mathsf{T}},\\\boldsymbol{b}&=(b_1,b_2,\cdots,b_q)^{\mathsf{T}}=\boldsymbol{x}_0+(\overline{\lambda}_1^*+\varepsilon_0,\overline{\lambda}_2^*+\varepsilon_0,\cdots,\overline{\lambda}_q^*+\varepsilon_0)^{\mathsf{T}}.\end{aligned}\tag{7}$$

则 $S_{[\boldsymbol{a},\boldsymbol{b}]}^{q-1}$ 是 $\mathscr{X}$ 的近似上下界覆盖.

第二阶段, 产生 $S_{[\boldsymbol{a},\boldsymbol{b}]}^{q-1}$ 上的稠密格点覆盖. 给定一个常数 $\delta_0>0$ 设为步长, 使得

$$a_i<a_i+\delta_0<a_i+2\delta_0<\cdots<a_i+k_i\delta_0\leqslant b_i<a_i+(k_i+1)\delta_0,\tag{8}$$

或者

$$b_i-(k_i+1)\delta_0<a_i\leqslant b_i-k_i\delta_0<a_i-(k_i-1)\delta_0<\cdots<b_i-\delta_0<b_i,$$

$i=1,2,\cdots,q$. 这里的 a_i, b_i 是 (7) 中的向量 $\boldsymbol{a}$, $\boldsymbol{b}$ 中的第 i 个元素. 以 (8) 为例, 这样可以将区间

$$[a_i,b_i]=\bigcup_{j=1}^{k_i}[a_i+(j-1)\delta_0,a_i+j\delta_0)\cup[a_i+k_i\delta_0,b_i]$$

划分为由 k_i 个等距区间与长度小于 δ_0 的闭区间相并而得. 特别的, 当 $a_i+k_i\delta_0=b_i$ 时, $[a_i,b_i]$ 被划分为 k_i 个等距区间.

定义一种矩阵运算, 对于 m 行 n 列的矩阵 A 和向量 $\boldsymbol{c}=(c_1,c_2,\cdots,c_m)^{\tau}$, 其中的元素 c_i 非 0 即 1, 若 $c_{i_1}=c_{i_2}=\cdots=c_{i_{m_1}}=1$, $c_{j_1}=c_{j_2}=\cdots=c_{j_{m_2}}=0$, $m_1+m_2=m$, 令

$$F(A\,|\,\boldsymbol{c}=1)=[e_{i_1},e_{i_2},\cdots,e_{i_{m_1}}]^{\tau}A, \tag{9}$$

其中 e_i 是 m 维列向量, 它的第 i 个元素为 1, 其余元素都为 0.

将 $S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$ 中各分量按变程大小重新排列. 不失一般性, 假设 $b_1-a_1\leqslant b_2-a_2\leqslant\cdots\leqslant b_q-a_q$, 使用以下算法迭代可生成 $S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$ 上的格点覆盖.

算法 1

输入步长 δ_0, 产生 q 个向量 $\boldsymbol{t}_i=(t_{i1},t_{i2},\cdots,t_{ik_i})^{\tau}$, 令 $T^{(1)}=\boldsymbol{t}_i$, 其中 $t_{ij}=a_i+(j-1)\delta_0$, $i=1,2,\cdots,q$, $j=1,2,\cdots,k_i+1$;

for $i=2$ to $q-1$ do

  $T'^{(i)}=[T^{(i-1)}\otimes\mathbf{1}_{k_i},\mathbf{1}_{k^{(i)}}\otimes\boldsymbol{t}_i]$, $k^{(i)}$ 为 $T^{(i-1)}\otimes\mathbf{1}_{k_i}$ 的行数,
  $n^{(i)}=k^{(i)}\times k_i$ 为 $T'^{(i)}$ 的行数;

  $\boldsymbol{d}^{(i)}=\mathbf{1}_{n^{(i)}}-T'^{(i)}\mathbf{1}_i=\big(d_1^{(i)},d_2^{(i)},\cdots,d_{n^{(i)}}^{(i)}\big)^{\tau}$;

  $\boldsymbol{c}^{(i)}=\big(c_1^{(i)},c_2^{(i)},\cdots,c_{n^{(i)}}^{(i)}\big)^{\tau}$, 其中 $c_j^{(i)}=I(0\leqslant d_j^{(i)}\leqslant 1)$, $j=1,2,\cdots,n^{(i)}$;

  $T^{(i)}=F(T'^{(i)}\,|\,\boldsymbol{c}^{(i)}=1)$;

end

最后再计算出

$$T'^{(q)}=[T^{(q-1)},\boldsymbol{t}'_q],$$

其中 $\boldsymbol{t}'_q=(t'_{q1},t'_{q2},\cdots,t'_{qn^{(q-1)}})^{\tau}=\mathbf{1}_{n^{(q-1)}}-T^{(q-1)}\mathbf{1}_{q-1}$, 再令 $\boldsymbol{c}^{(q)}=(c_1^{(q)},c_2^{(q)},\cdots,c_{n^{(q-1)}}^{(q)})^{\tau}$, 其中 $c_j^{(q)}=I(a_q\leqslant t'_{qj}\leqslant b_q)$, $j=1,2,\cdots,n^{(q-1)}$. 剔除不满足条件的格点, 将由算法 1 所生成的格子点矩阵令为

$$L^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}(\delta_0)=F(T'^{(q)}\,|\,\boldsymbol{c}^{(q)}=1), \tag{10}$$

并将各行格点构成的集合记为 $\mathscr{L}^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}(\delta_0)$, 从而得到覆盖 $S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$ 的格子点集当 q 较小时 (例如 $q\leqslant 5$), 可以直接执行以下矩阵生成方法, 令 $T'=(\boldsymbol{t}'_1,\boldsymbol{t}'_2,\cdots,\boldsymbol{t}'_q)$, 其中 $\boldsymbol{t}'_i=\mathbf{1}_{k_1k_2\cdots k_{i-1}}\otimes\boldsymbol{t}_i\otimes\mathbf{1}_{k_{i+1}k_{i+2}\cdots k_{q-1}}$, $i=1,2,\cdots,q-1$, $\boldsymbol{t}'_q=(t'_1,t'_2,\cdots,t'_K)^{\tau}=\mathbf{1}_K-(\boldsymbol{t}'_1,\boldsymbol{t}'_2,\cdots,\boldsymbol{t}'_{q-1})\mathbf{1}_{q-1}$, 这里 $K=\prod\limits_{j=1}^{q-1}k_j$. 记向量 $\boldsymbol{c}=(c_1,c_2,\cdots,c_K)^{\tau}$ 中的元素 $c_l=I(a_q\leqslant t'_l\leqslant b_q)$, $l=1,2,\cdots,K$, $I(\cdot)$ 是示性函数. 再令 $T=F(T'\,|\,\boldsymbol{c}=1)$ 中的各行即为覆盖 $S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$ 的格子点. 如果在 (10) 式中, 将每个分量进行划分的步长进行调整, 令

$$[a_i,b_i)=\bigcup_{j=1}^{k_i}[a_i+(j-1)\delta_i,a_i+j\delta_i],\qquad i=1,2,\cdots,q,$$

即通过调整 δ_0 为 δ_i, 使得 $b_i = a_i + j\delta_i$, 在算法 1 中 $\boldsymbol{t}_i = (t_{i1}, t_{i2}, \cdots, t_{ik_i})^{\mathsf{T}}$ 中, $T^{(1)} = \boldsymbol{t}_i$, 其中 $t_{ij} = a_i + (j-1)\delta_i$, $i = 1, 2, \cdots, q$, $j = 1, 2, \cdots, k_i + 1$, 记 $\Delta = (\delta_1, \delta_2, \cdots, \delta_q)^{\mathsf{T}}$, 由此产生的 $S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$ 上的格点覆盖记为 $\mathscr{L}^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}(\Delta)$. 一般的, 当步长取得越小, $S^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}$ 上的格点覆盖越稠密, 此时, 我们令

$$\mathscr{L}_{\mathscr{X}}(\delta_0) = \mathscr{L}^{q-1}_{[\boldsymbol{a},\boldsymbol{b}]}(\delta_0) \cap \mathscr{X}$$

就是 $\mathscr{X}$ 上的格点覆盖.

§4. 最优设计的检测与比较

构造试验域 $\mathscr{X}$ 上的稠密格子点集, 一方面可以用于检测一个设计 ξ 是否为最优设计, 另一方面可以比较若干个局部最优设计的优劣.

要判断一个设计是否为最优的, 在约束较为复杂的试验域内, 我们无法证明对于所有的 $\boldsymbol{x} \in \mathscr{X}$ 都满足最优性等价条件

$$\psi_\phi(\boldsymbol{x}; \xi) = \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{x}) \Psi_\phi(M(\xi)) \boldsymbol{f}(\boldsymbol{x}) \leqslant c_\phi, \tag{11}$$

但可以通过检测到存在 $\boldsymbol{x} \in \mathscr{U}$ 有 $\psi_\phi(\boldsymbol{x}; \xi) > c_\phi$ 就否定设计 ξ 的最优性, 这里 $\Psi_\phi(M(\xi))$ 与 c_ϕ 分别是 ϕ-最优准则下关于信息矩阵的函数与某个常数, 下标 ϕ 为 D-, A-, R-等. 例如 D-最优准则下的 $\Psi_D(M(\xi)) = M^{-1}(\xi)$, $c_D = m$, A-最优准则下 $\Psi_A(M(\xi)) = M^{-2}(\xi)$, $c_A = \mathrm{tr}(M^{-1}(\xi))$, 等. 令

$$h_\phi(\boldsymbol{x}; \xi) = \psi_\phi(\boldsymbol{x}; \xi) - c_\phi,$$

由于二次型函数 $h_\phi(\boldsymbol{x}; \xi)$ 在 S^{q-1} 上是连续的, 如果一个设计 ξ 不是最优的, 则意味着存在若干子区域 $\mathscr{U}_i$, $i = 1, 2, \cdots, k$, 使得对于任意的 $\boldsymbol{x} \in \bigcup\limits_{i=1}^{k} \mathscr{U}_i$ 都有 $h_\phi(\boldsymbol{x}; \xi) > 0$. 不失一般性, 假设 $\mathscr{U}_i$ 都是 $\mathscr{X}$ 内没有共同内点的子单纯形, 它的 q 个顶点分别为 $\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \cdots, \boldsymbol{x}_{iq}$, 并记 $U_i^{\mathsf{T}} = [\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \cdots, \boldsymbol{x}_{iq}]$, $i = 1, 2, \cdots, k$. 则有

$$\mathrm{Vol}(\mathscr{U}_i) = \frac{\sqrt{q}}{(q-1)!} \det(U_i), \qquad i = 1, 2, \cdots, k.$$

所以, 只要 $\sum\limits_{i=1}^{k} \det(U_i) \neq 0$, 采用 $\mathscr{X}$ 上的稠密格子点集 $\mathscr{L}_{\mathscr{X}}(\delta_0)$ 来检测设计 ξ 是否为最优设计, 必然有

$$\lim_{\delta_0 \to 0} \mathsf{P}\Big(\bigcup_{\boldsymbol{x} \in \mathscr{L}_{\mathscr{X}}(\delta_0)} \{h_\phi(\boldsymbol{x}; \xi) > 0\} \neq \emptyset \Big) = 1.$$

也就是说, 只要格子点集足够稠密, 检测到不满足最优性等价条件的区域的概率为 1. 即随着格点阶数的增大, 检测到非最优设计区域的概率趋近于 1.

以 3 分量非最优设计为例, 如图 1 中的是函数 $h_\phi(\boldsymbol{x};\xi)$ 的曲面图. 当 q 很大时, 采用稠密格点的数量十分巨大, 为了检测到不满足最优性的区域, 可以考虑在 $\mathscr{X}$ 内使用随机搜索算法 [10] 来搜索 $h_\phi(\boldsymbol{x};\xi)$ 在 $\mathscr{X}$ 的极大值, 如果存在 $h_\phi(\boldsymbol{x};\xi)>0$, 则否定设计 ξ 的最优性. 不论使用格点检测或是随机搜索, 目的都是确定函数 $h_\phi(\boldsymbol{x};\xi)$ 的极大值, 如果曲面全部位于 0 下方, 则肯定设计的最优性, 反之, 存在函数 $h_\phi(\boldsymbol{x};\xi)>0$ 的部分, 则否定设计的最优性. 由此可见, 虽然我们无法证明一个设计在 $\mathscr{X}$ 上的最优性, 但可以通过检测到 $h_\phi(\boldsymbol{x};\xi)>0$ 的区域来否定它的最优性. 令

$$p(\xi;\delta_0)=\frac{\sum\limits_{\boldsymbol{x}\in\mathscr{L}_{\mathscr{X}}(\delta_0)} I(h_\phi(\boldsymbol{x};\xi)>0)}{\#\{\mathscr{L}_{\mathscr{X}}(\delta_0)\}}, \tag{12}$$

从另一个角度衡量了设计的优劣. 直观上可见, 如果 $p(\xi;\delta_0)$ 越接近于 0, 说明近似最优的效果越好, 反之则越差.

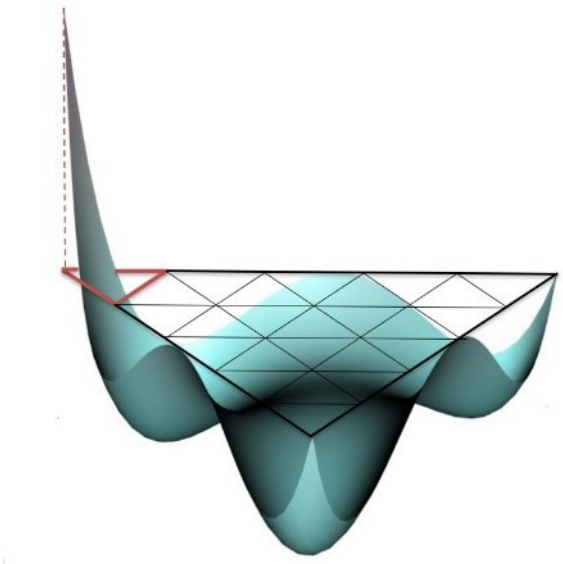

图 1　非最优设计的格点检测示意

很多情形下, 由于区域约束复杂, 试验域 $\mathscr{X}$ 极不规则, 我们很难求解得 $\mathscr{X}$ 上的全局最优设计. 使用优化算法受到初值的影响, 搜索得出一系列局部最优解都不相同. 这些局部最优设计有可能试验点的分布却完全不同, 但在某种最优准则下的表现差别并不明显. 比如两个局部最优设计 ξ_1 与 ξ_2 的信息矩阵的行列式值近乎相等, 即 $\det(M(\xi_1))\approx\det(M(\xi_2))$, 此时应如何选择最终结果就是一个比较棘手的问题. 根据 I-最优设计的思想, 除了比较这些局部最优设计的最优性度量函数 (如 $\det(M(\xi))$, $\mathrm{tr}(M^{-1}(\xi))$ 等) 之外, 还应对设计的整体预测效果进行比较. 因为模型 (1) 在 $\boldsymbol{x}\in\mathscr{X}$ 上预测值的方差为 $\mathsf{Var}\,(\widehat{y}(\boldsymbol{x}))=\psi_D(\boldsymbol{x},\xi)$.

I-最优设计是使得预测值方差的均值达到最小的一种设计, 即

$$\mathsf{E}[\mathsf{Var}\,(\widehat{y}(\boldsymbol{x}))] = \Big(\int_{\mathscr{X}} \mathrm{d}\boldsymbol{x}\Big)^{-1} \int_{\mathscr{X}} \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{x}) M^{-1}(\xi) \boldsymbol{f}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \to \inf_{\zeta \in \Xi}.$$

如果 $\mathscr{X}$ 上的格点覆盖 $\mathscr{L}_{\mathscr{X}}(\delta_0)$ 的步长越小, 则意味着

$$\lim_{\delta_0 \to 0} \frac{1}{\#\{\mathscr{L}_{\mathscr{X}}(\delta_0)\}} \sum_{\boldsymbol{x} \in \mathscr{L}_{\mathscr{X}}(\delta_0)} \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{x}) M^{-1}(\xi) \boldsymbol{f}(\boldsymbol{x}) = \mathsf{E}[\mathsf{Var}\,(\widehat{y}(\boldsymbol{x}))].$$

所以, 估计量

$$\widehat{\sigma}(\xi; \delta_0) = \frac{1}{\#\{\mathscr{L}_{\mathscr{X}}(\delta_0)\}} \sum_{\boldsymbol{x} \in \mathscr{L}_{\mathscr{X}}(\delta_0)} \psi_D(\boldsymbol{x}, \xi) \tag{13}$$

可作为 $\mathsf{E}[\mathsf{Var}\,(\widehat{y}(\boldsymbol{x}))]$ 的一个相合估计.

如果要比较 p 个设计 ξ_i, $i = 1, 2, \cdots, p$ 的最优性, 令

$$v_{ij} = \psi_D(\boldsymbol{x}_j; \xi_i), \qquad i = 1, 2, \cdots, p,\ j = 1, 2, \cdots, N, \tag{14}$$

其中 $N = \#\{\mathscr{L}\mathscr{X}(\delta_0)\}$. 这样不仅可以比较出 p 组数据均值之间的差异, 还可以观察每组数据的分布情况, 进而更好的做出判断.

§5. 局部最优设计的生成

当 $\mathscr{L}_{\mathscr{X}}(\delta_0)$ 在 $\mathscr{X}$ 上稠密覆盖时, 最优设计的点集极有可能包含于 $\mathscr{L}_{\mathscr{X}}(\delta_0)$ 中. 所以我们可以用 $\mathscr{L}_{\mathscr{X}}(\delta_0)$ 作为备选点集, 然后从中抽取出尽可能达到最优设计的点集. 以 D-最优设计为例, 按照以下步骤进行搜索.

首先, 从 $\mathscr{L}_{\mathscr{X}}(\delta_0) = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ 中选出一个子集 $\{\boldsymbol{x}_1^{(0)}, \boldsymbol{x}_2^{(0)}, \cdots, \boldsymbol{x}_k^{(0)}\}$ 作为初值. 然后将 $\boldsymbol{x}_1^{(0)}, \boldsymbol{x}_2^{(0)}, \cdots, \boldsymbol{x}_k^{(0)}$ 按照以下方式依次逐个进行迭代, 令

$$\boldsymbol{x}_i^{(j)} = \arg \max_{\boldsymbol{x}_i^{(j-1)} \in \mathscr{L}_{\mathscr{X}}(\delta_0)} \det \Big[\frac{1}{k} \sum_{l=1}^{i-1} \boldsymbol{f}\big(\boldsymbol{x}_l^{(j)}\big) \boldsymbol{f}^{\mathsf{T}}\big(\boldsymbol{x}_l^{(j)}\big) + \frac{1}{k} \sum_{l=i}^{k} \boldsymbol{f}\big(\boldsymbol{x}_l^{(j-1)}\big) \boldsymbol{f}^{\mathsf{T}}\big(\boldsymbol{x}_l^{(j-1)}\big)\Big],$$

其中 $i = 1, 2, \cdots, k$, $j = 1, 2, \cdots, a$. 依次迭代完 a 轮后得到点集 $\{\boldsymbol{x}_1^{(a)}, \boldsymbol{x}_2^{(a)}, \cdots, \boldsymbol{x}_k^{(a)}\}$. 如果现阶段的局部最优设计预测效果仍不佳, 可以考虑在此基础上添加试验点. 令

$$\boldsymbol{x}_{k+1}^{(a)} = \arg \max_{\boldsymbol{x} \in \mathscr{L}_{\mathscr{X}}(\delta_0)} \det \Big[\frac{1}{k+1} \sum_{l=1}^{i-1} \boldsymbol{f}\big(\boldsymbol{x}_l^{(a)}\big) \boldsymbol{f}^{\mathsf{T}}\big(\boldsymbol{x}_l^{(a)}\big) + \frac{1}{k+1} \sum_{l=i}^{k} \boldsymbol{f}(\boldsymbol{x}) \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{x})\Big],$$

当添加的试验点与原有的试验点重复时, 则无需再进行添加.

最后, 对于添加了若干个试验点后的 p 点设计, 再使用测度搜索算法 [12], 求得

$$\boldsymbol{w}^* = \arg \max_{\boldsymbol{w} \in S^{p-1}} \det \Big[\sum_{l=1}^{p} w_l \boldsymbol{f}\big(\boldsymbol{x}_l^{(a)}\big) \boldsymbol{f}^{\mathsf{T}}\big(\boldsymbol{x}_l^{(a)}\big)\Big],$$

这样就得到一个局部最优设计

$$\xi^* = \begin{pmatrix} \boldsymbol{x}_1^{(a)} & \boldsymbol{x}_2^{(a)} & \cdots & \boldsymbol{x}_p^{(a)} \\ w_1^* & w_2^* & \cdots & w_p^* \end{pmatrix},$$

其中 $\boldsymbol{w}^* = (w_1^*, w_2^*, \cdots, w_p^*)^{\mathsf{T}}$.

在约束复杂的情形下, 我们很难获得全局最优的设计, 但按照这种方法可以得到一系列的局部最优设计. 然后在此基础上进行筛选, 最终所得到的设计是具有很大参考价值的.

§6. 模拟实例

为直观表现格点覆盖情况, 我们讨论 3 分量的混料规范多项式模型

$$\mathsf{E}(y) = \sum_{i=1}^{3} \beta_i x_i + \sum_{i<j}^{3} \beta_{ij} x_i x_j + \sum_{i<j}^{3} \gamma_{ij} x_i x_j (x_i - x_j) + x_1 x_2 x_3$$

在试验域 $\mathscr{X} = \{\boldsymbol{x} = (x_1, x_2, x_3) : \boldsymbol{x} \in S^{q-1},\ x_1^2 + x_2^2 \leqslant 0.49,\ x_2^2 + x_3^2 \geqslant 0.16\}$ 上的最优设计检测问题. 可见, $\mathscr{X}$ 是一个不规则的非凸区域, 在有的边界是不存在极端顶点的. 使用步长 $\delta_0 = 0.05$ 产生试验域上的格点覆盖, 试验区域及格点覆盖如图 2 所示.

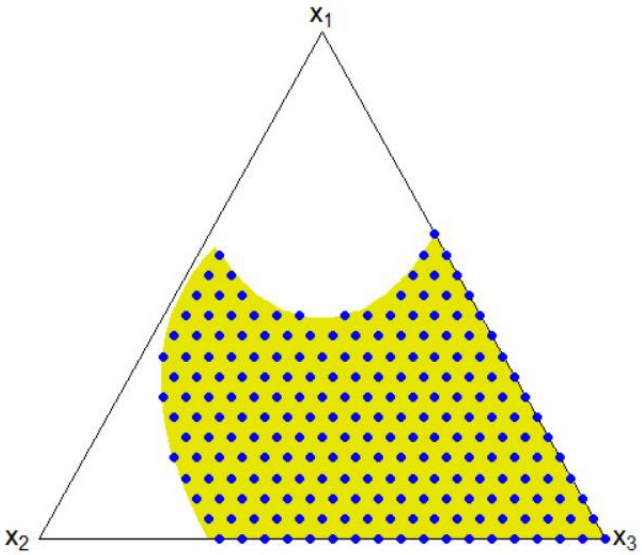

图 2 覆盖约束区域

下面, 我们使用步长 $\delta_0 = 0.005$ 产生试验域上的格点覆盖 $\mathscr{L}_{\mathscr{X}}(\delta_0)$, 并从中选取不同初值, 在保证 D-最优的前提下, 按照第 5 节的方法, 选择不同初值并搜索产生 10 个设计. 对这 10 个设计中的设计点进行编号, 如表 1 所示.

表 1　D-最优设计点集

编号	x_1	x_2	x_3	编号	x_1	x_2	x_3	编号	x_1	x_2	x_3
1	0.000	0.000	1.000	17	0.095	0.500	0.405	33	0.295	0.000	0.705
2	0.000	0.195	0.805	18	0.125	0.165	0.710	34	0.305	0.630	0.065
3	0.000	0.200	0.800	19	0.140	0.685	0.175	35	0.350	0.295	0.355
4	0.000	0.215	0.785	20	0.155	0.475	0.370	36	0.360	0.600	0.040
5	0.000	0.225	0.775	21	0.160	0.000	0.840	37	0.395	0.000	0.605
6	0.000	0.430	0.570	22	0.160	0.450	0.390	38	0.405	0.570	0.025
7	0.000	0.435	0.565	23	0.165	0.000	0.835	39	0.435	0.000	0.565
8	0.000	0.440	0.560	24	0.165	0.680	0.155	40	0.435	0.230	0.335
9	0.000	0.490	0.510	25	0.170	0.520	0.310	41	0.440	0.000	0.560
10	0.000	0.505	0.495	26	0.175	0.500	0.325	42	0.440	0.225	0.335
11	0.000	0.660	0.340	27	0.205	0.000	0.795	43	0.440	0.240	0.320
12	0.000	0.700	0.300	28	0.240	0.000	0.760	44	0.445	0.220	0.335
13	0.080	0.170	0.750	29	0.245	0.000	0.755	45	0.450	0.205	0.345
14	0.085	0.170	0.745	30	0.275	0.000	0.725	46	0.570	0.405	0.025
15	0.090	0.165	0.745	31	0.285	0.000	0.715	47	0.600	0.000	0.400
16	0.090	0.500	0.410	32	0.290	0.000	0.710				

在这 10 个设计中, $\xi_1, \xi_2, \cdots, \xi_8$ 是等测度的饱和设计. ξ_9 和 ξ_{10} 都是在原有的基础上添加了 4 个试验点, 其中 ξ_9 为等测度设计, ξ_{10} 则通过搜索算法得到的测度向量解为 $\boldsymbol{w}^{\mathrm{T}} = (0.038, 0.05, 0.097, 0.096, 0.096, 0.074, 0.083, 0.093, 0.093, 0.077, 0.048, 0.057, 0.049, 0.049)$.

将这 10 个设计的点集对应表 1 的编号列与表 2 的第 2 列. 并计算出

$$\Phi_D(\xi) = 10^{27} \times \det M(\xi_i), \quad \Phi_A(\xi) = \ln\left(\mathrm{tr} M^{-1}(\xi_i)\right), \quad \Phi_R(\xi) = \sum_{j=1}^{p} \ln\left(M^{-1}(\xi_i)\right)_{jj},$$

$i = 1, 2, \cdots, 10$, 置于表 2 的第 3, 4, 5 列. 由 (14) 式计算得出的 10 组数据 v_{ij}, $i = 1, 2, \cdots, 10$, $j = 1, 2, \cdots, N$. 通过 (12) 式计算得到各设计的 $p(\xi_i; \delta_0)$ 值置于表 2 的第 6 列, 再由 (13) 求出这 10 组数据的均值 $\widehat{\sigma}(\xi_i; \delta_0)$ 置于表 2 的第 7 列.

更进一步, 直观地比较这 10 个设计的优劣, 将 10 组数据减去 10, 绘制成箱图, 如图 3 所示.

由表 2 可见, 仅从行列式大小来看, ξ_9 的表现最优, 但是 $p(\xi_9; \delta_0) = 0.21$ 不如 $p(\xi_{10}; \delta_0) = 0.079$ 的表现好. 并且 ξ_9 与 ξ_{10} 的行列式值差异也不大, 预测方差估计量 $\widehat{\sigma}(\xi_9; \delta_0) = 8.8$ 不如 $\widehat{\sigma}(\xi_{10}; \delta_0) = 8.479$ 小. 虽然 $p(\xi_1; \delta_0) = 0.078$ 以及 $\sigma(\xi_1; \delta_0) = 8.274$ 都表现良好, 但 ξ_1 的行列式值并不理想, 与 ξ_9, ξ_{10} 相差较大. 在保证 D-最优准则的前提下, 其他的设计相比起来都各自有所不足. 因而我们形成的初步判断是, 在非饱和设计的情形下, ξ_{10} 是综合表现相比其他设计是最好的. 在饱和设计的情形下, 应首选 ξ_1. 由图 3 可见, ξ_{10} 的异常

表 2　10 个设计的设计点编号及最优性比较

设计	设计点编号	$\Phi_D(\xi)$	$\Phi_A(\xi)$	$\Phi_R(\xi)$	$p(\xi_i;\delta_0)$	$\widehat{\sigma}(\xi_i;\delta_0)$
ξ_1	1,6,12,15,25,33,36,44,46,47	3.35	12.088	85.369	0.078	8.274
ξ_2	1,2,10,12,23,34,35,39,46,47	2.56	12.310	87.345	0.643	12.002
ξ_3	1,5,11,19,20,38,39,45,46,47	2.75	12.244	86.787	0.224	9.351
ξ_4	1,2,10,12,22,30,36,44,46,47	2.70	12.648	88.868	0.393	10.673
ξ_5	1,4,12,17,24,28,38,42,46,47	2.93	12.442	87.527	0.377	10.035
ξ_6	1,8,12,13,25,32,36,44,46,47	3.32	12.126	85.510	0.090	8.346
ξ_7	1,7,12,14,25,31,36,44,46,47	3.33	12.125	85.543	0.083	8.314
ξ_8	1,4,12,16,24,28,38,40,46,47	2.93	12.437	87.498	0.380	10.038
ξ_9	1,3,9,12,18,24,26,27,36,37,38,43,46,47	4.85	11.828	83.686	0.210	8.800
ξ_{10}	1,2,4,7,12,14,21,24,25,31,36,41,46,47	4.72	11.914	84.071	0.079	8.479

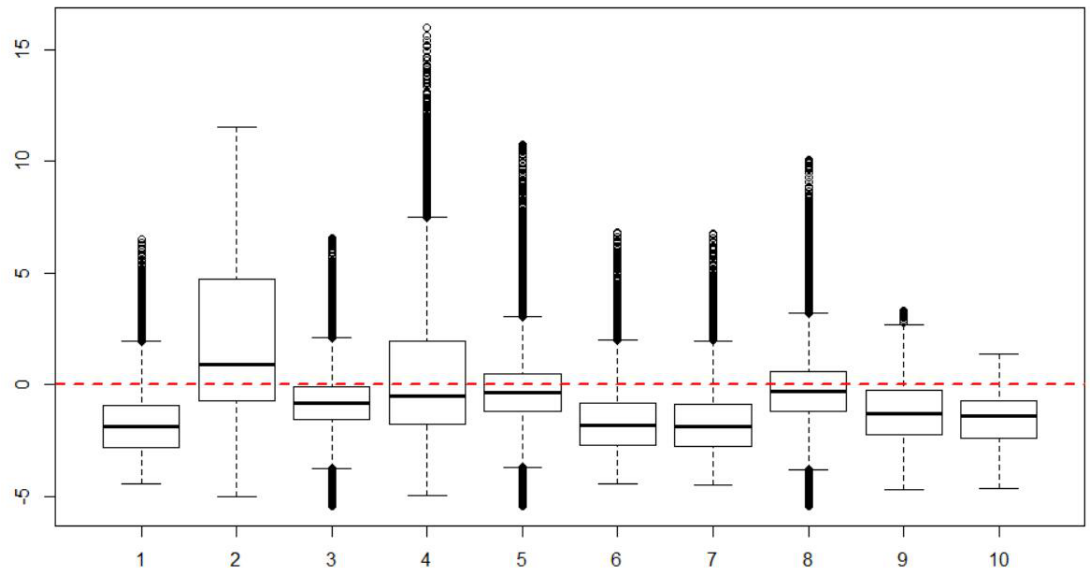

图 3　10 个设计的预测方差数据比较

值较少, 并且箱体大部分位于 0 下方, 在非全局最优设计类中, ξ_{10} 是比较好的一个设计.

§7. 结　　论

本文通过构造复杂约束区域上的稠密格点覆盖, 研究了试验域上的设计最优性的检测, 局部最优设计优劣性的比较以及近似最优设计的生成算法. 总体而言, 只要能产生约束试验域内足够稠密的格子点集, 对于设计的评价是很有帮助的.

要生成含有附加约束的试验域内的稠密覆盖格子点集, 一般是将无附加约束的单纯形内的高阶格子点集剔除掉约束以外的部分即得到. 而如果约束域区域很小, 要剔除约束以外的部分必然事倍功半. 这就需要构造出一些特殊子区域的格子点集构造算法, 并使用下界约束变换将其映射到小的子区域内. 但对于高维复杂约束的混料问题仍然存在很大的困难. 一方面, 要产生约束域上的稠密格点覆盖, 由于维数高, 所生成的点的个数十分巨大. 例如, R 中的 AlgDesign 包中产生 10 分量 20 阶格子点集已经到达了极限, mixexp 包中的 XVERT 函数至多只能计算 12 个分量的极端顶点. 另一方面, 对于复杂约束域, 首先就需要确定该区域的上下界约束, 然而这一步工作都比较困难的. 为此, 我们设想可以在约束域内产生若干随机点, 然后建立搜索边界的目标函数, 进而搜索出试验域的上下界约束向量. 关于这一问题仍有待进一步研究.

目前, R 语言中关于混料试验的程序包能生成多分量低阶格子点集. 由于高维高阶格子点集是一个巨大的稀疏矩阵. 研究生成高阶格子点集的算法, 并考虑在对称区域上, 对高阶格子点集进行稀疏分解. 对于一些特殊的对称设计, 基于格子点集提出更为简化的均匀性度量准则. 李光辉[13] 中提出了格子点集的对称表示法, 对于 q 分量的高阶格子点集, 使用稀疏分解和对称表示能使得储存空间大大节省, 作为 NT-net 用于度量设计的均匀性时也更为高效.

参 考 文 献

[1] ELFVING G. Optimum allocation in linear regression theory [J]. *Ann Math Statistics*, 1952, **23(2)**: 255–262.

[2] MCLEAN R A, ANDERSON V L. Extreme vertices design of mixture experiments [J]. *Technometrics*, 1966, **8(3)**: 447–454.

[3] SNEE R D, MARQUARDT D W. Extreme vertices designs for linear mixture models [J]. *Technometrics*, 1974, **16(3)**: 399–408.

[4] SNEE R D. Experimental designs for mixture systems with multicomponent constraints [J]. *Comm Statist Theory Methods*, 1979, **8(4)**: 303–326.

[5] SNEE R D. Developing blending models for gasoline and other mixtures [J]. *Technometrics*, 1981, **23(2)**: 119–130.

[6] 关颖男. 混料凸多面体顶点的分枝构造法及修正的 CONSIM 算法 [J]. *应用数学学报*, 1984, **7(3)**: 340–346.

[7] KARUNANITHI A T, ACHENIE L E K, GANI R. Optimal (Solvent) mixture design through a decomposition based CAMD methodology [J]. *Comput Aided Chem Eng*, 2004, **18**: 217–222.

[8] LIMMUN W, BORKOWSKI J J, CHOMTEE B. Using a genetic algorithm to generate D-optimal designs for mixture experiments [J]. *Qual Reliab Engng Int*, 2013, **29(7)**: 1055–1068.

[9] WONG W K, CHEN R B, HUANG C C, et al. A modified particle swarm optimization technique for finding optimal designs for mixture models [J]. *PLoS One*, 2015, **10(6)**: e0124720.

[10] LI G H, ZHANG C Q. Random search algorithm for optimal mixture experimental design [J]. *Comm Statist Theory Methods*, 2018, **47(6)**: 1413–1422.

[11] KIEFER J. Optimal design: variation in structure and performance under change of criterion [J]. *Biometrika*, 1975, **62(2)**: 277–288.

[12] 李光辉, 张崇岐. 具有复杂约束混料试验的渐近 D-最优设计 [J]. 应用概率统计, 2017, **33(2)**: 203–220.

[13] 李光辉. 混料试验的拟分量变换设计的理论及应用 [D]. 广州: 广州大学, 2018.

Lattice Evaluation of the Optimal Mixture Design in the Complex Constraint Region

LI Guanghui

(*School of Science, Kaili University, Kaili, 556011, China*)

LI Junpeng ZHANG Chongqi

(*School of Economic and Statistic, Guangzhou University, Guangzhou, 510006, China*)

Abstract: It is difficult to prove the optimality of a mixture design in the region with complex constraints, however, it is relatively easy to negate the optimality of the design. In this paper, we construct a set of dense lattice points in the experimental region with complex constraint. It can not only be used to evaluate the optimization of a design, but also to compare the advantages of multiple local optimal designs. As an alternative set of points, the densely covered set of lattice points can produce the local optimal design in the constrained region. This method is effective through case analysis.

Keywords: mixture experiment; optimal design; lattice point set

2020 Mathematics Subject Classification: 62K05; 62K99