

11-17-2020

A Fast Latin Hyper Cube Experiment Design Method Based on Sudoku Grouping

Tiantian Zhang

1. School of Automation Science, Beihang University (BUAA), Beijing 100191, China; ;

Li Ni

1. School of Automation Science, Beihang University (BUAA), Beijing 100191, China; ;

Guanghong Gong

1. School of Automation Science, Beihang University (BUAA), Beijing 100191, China; ;

Yuanjie Lu

2. Chinese Aeronautical Establishment, Beijing 100129, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

A Fast Latin Hyper Cube Experiment Design Method Based on Sudoku Grouping

Abstract

Abstract: In order to solve the long optimization time and parameter level combination explosion in the complex experiment design space of Latin Hypercube design, which is one of the most popular method in experiment design, *a fast Sudoku grouping-based method is proposed*. The optimal seed design is expanded and transformed in the grouped spaces. Experiments are conducted to compare the Sudoku grouping-based Latin Hypercube design method with other two commonly used Latin Hypercube design methods in the middle and high dimension experiment space. The results show that Sudoku grouping method is more efficient in computation and has better space-filling performance.

Keywords

experiment design, latin hypercube design, sudoku grouping, extended translational propagating

Recommended Citation

Zhang Tiantian, Li Ni, Gong Guanghong, Lu Yuanjie. A Fast Latin Hyper Cube Experiment Design Method Based on Sudoku Grouping[J]. Journal of System Simulation, 2020, 32(11): 2185-2191.

一种基于数独分组的拉丁超立方试验设计方法

张甜甜¹, 李妮¹, 龚光红¹, 卢元杰²

(1. 自动化科学与电气工程学院, 北京航空航天大学, 北京 100191; 2. 中国航空研究院, 北京 100029)

摘要:为解决拉丁超立方设计在复杂因子的实验设计空间中存在优化时间长和因子水平组合爆炸的问题。提出一种基于数独分组的拉丁超立方设计方法。该方法对因子通过数独分组降维, 在降维后的各个子空间中平移种子设计。将基于数独分组的拉丁超立方设计与另外两种常用的拉丁超立方设计方法在中高维度试验空间中进行对比, 结果表明, 基于数独分组的拉丁超立方设计有较优的空间填充性和最短的运行时间。

关键词: 试验设计; 拉丁超立方设计; 数独分组; 扩展平移传播

中图分类号: TP391.9 文献标识码: A 文章编号: 1004-731X (2020) 11-2185-07

DOI: 10.16182/j.issn1004731x.joss.20-FZ0472

A Fast Latin Hyper Cube Experiment Design Method Based on Sudoku Grouping

Zhang Tiantian¹, Li Ni¹, Gong Guanghong¹, Lu Yuanjie²

(1. School of Automation Science, Beihang University (BAA), Beijing 100191, China;

2. Chinese Aeronautical Establishment, Beijing 100129, China)

Abstract: In order to solve the long optimization time and parameter level combination explosion in the complex experiment design space of Latin Hypercube design, which is one of the most popular method in experiment design, a fast Sudoku grouping-based method is proposed. The optimal seed design is expanded and transformed in the grouped spaces. Experiments are conducted to compare the Sudoku grouping-based Latin Hypercube design method with other two commonly used Latin Hypercube design methods in the middle and high dimension experiment space. The results show that Sudoku grouping method is more efficient in computation and has better space-filling performance.

Keywords: experiment design; latin hypercube design; sudoku grouping; extended translational propagating

引言

面向仿真的实验设计(Design of Experiment, DOE)通过改变仿真系统的输入, 以获得不同输出, 进一步分析仿真系统输入与输出之间的关系, 以较少的实验次数尽可能地取得代表性较高的实验结

果。在复杂仿真系统中, 因子数众多, 需要进行的试验次数随之呈指数增长, 面临因子水平组合爆炸的问题。因此, 在 DOE 领域中, 需要在仿真空间中经济与科学地安排试验执行次数及分布位置^[1], 其中 McKay 等^[2]和 Iman 等^[3]提出的拉丁超立方设计(Latin Hypercube Design, LHD)是较为常用的方法。LHD 生成方法简单, 但生成的试验次数随因子水平数固定, 并且每次生成试验结果的空间分布性能随机, 需要对 LHD 设计结果进行优化。

以试验点集的空间填充性为设计方法的性能



收稿日期: 2020-07-10 修回日期: 2020-08-05;
作者简介: 张甜甜(1996-), 女, 山东菏泽, 硕士生, 研究方向为同时定位与建图、深度学习; 李妮(1980-), 女, 湖南衡阳, 博士, 教授, 研究方向为虚拟样机、混合现实技术; 龚光红(1968-), 女, 土家族, 四川石柱, 博士, 教授, 研究方向为系统仿真与虚拟现实。

<http://www.china-simulation.com>

• 2185 •

指标, 对 LHD 进行优化, 优化算法可分为两类。一类基于全局搜索算法, 通过选择不同的空间填充性标准, 对所有可能的拉丁超立方进行比较, 以得到最优的设计^[4], 但其搜索空间较大。另一类方法是利用机器学习算法^[5-8]对随机生成的 LHD 进行优化, 能够得到次优的设计, 但算法收敛速度慢。综上, LHD 优化算法需要较长运行时间获得一个相对较好的设计结果。

为快速生成近似最优的 LHD, Felipe Viana 等提出了一种基于平移传播算法的拉丁超立方设计算法 (Translational Propagation Latin Hypercube Design, TPLHD)^[9]。在此基础上, 翟刚等提出一种扩展的平移传播拉丁超立方设计方法 (Extended Translational Propagation LHD, ETPLHD)^[10], 对实验空间进行分层设计 TPLHD, 使得生成的试验点数可变。这 2 种方法均以二维的最优拉丁超立方设计为种子模板进行空间上的平移, 算法在因子数小于六维的低维空间简单高效, 但在中高维空间中分布集中, 填充性能较差; 并且上述 2 种算法的复杂度随输入因子个数呈指数上升, 在复杂仿真试验设计中, 常造成计算机的存储空间溢出。

为了解决上述问题, 本文设计了一种基于数独分组的扩展平移拉丁超立方设计方法 (Sudoku Grouping based ETPLHD, SDETPLHD), 该方法在中高维度下也能保持算法的快速性和较优的空间填充性。

1 ETPLHD 算法回顾

ETPLHD 算法对于 $n_p \times n_v$ 的试验空间, 首先选择某一维划分为 d 层, 然后对每一层进行 $n_p \times (n_v - 1)$ 的 ETPLHD 设计, 重复这一过程直到某一维设置采用 TPLHD 设计。以 16 水平 \times 3 因子的试验空间为例, 3 个因子维度分别记为 d^1 , d^2 , d^3 。选择 d^1 维分为 $d=16^{1/2}=4$ 层, 对每一层进行 d^2 和 d^3 维度上的 16×2 的 TPLHD 设计。最终, 共生成 $4 \times 16 = 64$ 个试验样本点, 如图 1 所示。

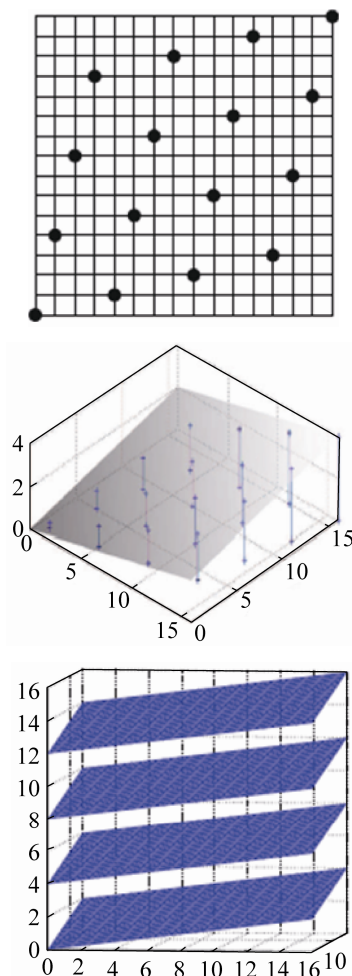
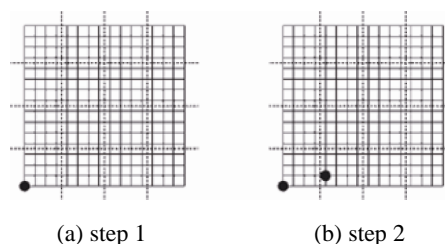


图 1 ETPLHD 的分层与平移
Fig. 1 Division and translation process of ETPLHD

16 水平 \times 2 因子的 TPLHD 设计算法步骤如图 2 所示。首先将包含各因子最低水平的小块作为第一个小块, 并放入种子设计, 如图 2(a)所示。然后任取一维, 将种子设计沿该维进行平移, 依次放入后续小块中, 如图 2(b)所示。重复平移步骤, 直到该维的所有小块都包含种子设计, 如图 2(d)所示。将这一维的试验点作为新的种子设计, 沿着其他维顺次重复以上过程, 直到试验空间中所有维都被填满, 如图 2(g)所示。



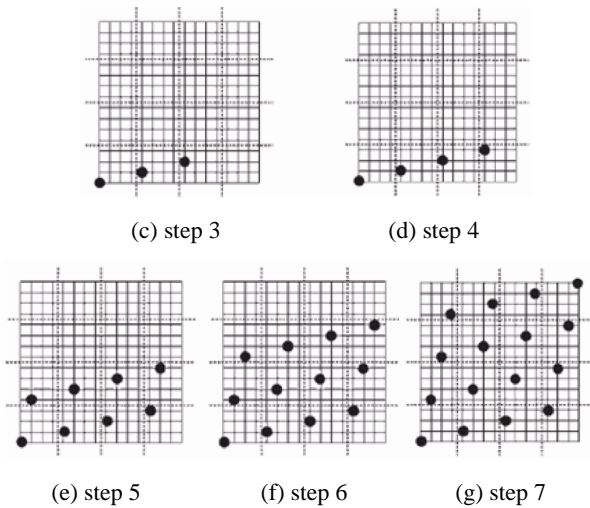


图 2 平移传播过程
Fig. 2 Translated Propagation Process

2 SDETPLHD 算法介绍

数独矩阵是一种特殊的拉丁超立方, 保持行和列的数字分布的均匀性。本节对 SDETPLHD 算法进行说明, 以 5 水平×13 因子的试验空间为例。假设共进行 $k=4$ 次分组, SDETPLHD 算法步骤如下:

step 1: 判断因子数 $n'_v = n_v = 13$ 是否能被 x 整除, $x \in \{Z^+ | 6 \leq x \leq 9\}$ 。如果能被整除, 将 n_v/x 记为单次分组数 n ; 如果不能被整除, 因子数加一并赋值给 n'_v , 重复上述操作。最终取 $n'_v = 15, n = 3$ 。

step 2: 生成 $n'_v/n = 5$ 因子, $n_p=55$ 水平的 ETPLHD。

step 3: 对 n'_v 上取到最接近的完全平方数 4^2 , 并生成 16 宫格的数独。

step 4: 对 16 宫格数独进行分组。如图 3 所示, 去掉第一行中编号大于 15 的因子, 对剩余因子均分为 $n=3$ 组; 每组因子按第(2)步的 ETPLHD 输出设计点集合, 其中序号大于 13 的因子不进行输出。取下一行, 重复上述步骤共 $k=4$ 次。

step 5: 将每次 3 组设计点集按因子编号升序串行排列, 作为一次试验设计结果; 将 4 次设计点集并行排列, 作为最终的试验设计结果。

SDETPLHD 算法的伪代码如下:

Algorithm SDETPLHD

Input:

k : the number of times to divide groups;
 np : the number of the design variables (dimensions) of the experiment space;
 nv : the maximum of all design variables levels.
 Output: result: the result of the experiment design.

```

1: Array result, D;
2: np'= np;
3: 11: If (np' % (4:6) == 0) n = np' / (4:6);
5:   else np' = np ++; goto 11;
8: Array seed = CreateETPLHD (np'/n, nv);
9: Sqrt (m) = round (np');
10: Array sudoku = CreateSudoku(Sqrt (m ));
11: for (times = 1: k)
12:   Array s = deleteExtraDimensions(sudoku
[times]);
13:   for (groups = 1: n)
14:     Mapping (s [groups], seed);
15:     .append( SerialArrangement( s[ : ] ));
16: result = ParallelArrangement(D[ : ] );
17: return result
    
```



图 3 数独分组过程
Fig. 3 Grouping process of Sudoku

3 LHD 算法性能对比实验

3.1 LHD 算法的空间填充性指标

验证试验设计点充满空间性能最常用的指标是 Φ_p 判据^[11], Φ_p 值越小, 说明点对之间的最小距离越大, 即在相同实验空间内, 具有相同试验点数条件下, 设计点集的空间填充性越好。 Φ_p 计算公式为:

$$\Phi_p = \left[\sum_{i=1}^n \sum_{j=i+1}^n d_{ij}^{-p} \right]^{1/p}$$

式中: p 为预先选定的整数; d_{ij} 为点对 x_i, x_j 之间的距离, $d_{ij} = d(x_i, x_j) = \left[\sum_{k=1}^n |x_{ik} - x_{jk}|^t \right]^{1/t}$, 通常按经验^[5]取值 $p=50, t=1$ 。

3.2 t-SNE 降维算法

为了更加直观的观察和比较不同试验设计算法在中高维空间中设计点的分布, 本文采用 t-SNE 算法^[12-13]进行 LHD 算法设计点集的降维可视化处理, 即将 ETPLHD 与 SDETPLHD 算法在 6~14 维空间的设计点映射为二维数据, 再进行可视化处理。

3.3 3 种 LHD 算法的实验配置与运行环境

本文依据运行时间、 Φ_p 判据, 对 3 种 LHD 算

法进行对比实验, 算法分别为:

(1) ETPLHD, 如 1.1 节所示。

(2) SDETPLHD, 如 1.2 节所示。

(3) lhsDesign, 这种算法是 Matlab 软件自带的设计。lhsDesign 算法原理是随机生成 200 个拉丁超立方设计, 根据点对之间最小距离最大化的标准来选出最终设计结果。

实验因子数为 6, 8, 10, 12, 14 这 5 种情况, 每种因子配置 16, 25, 32 这 3 种水平。实验运行的处理器为 Intel Core i5-3470 CPU at 3.20 GHz, 内存为 4 GB of RAM, 操作系统为 Win10, 软件包括 Visual Studio 2017, MATLAB R2014b。

4 实验结果分析

4.1 空间填充性与运行时间

4.1.1 分组次数 k 对 SDETPLHD 空间填充性能影响

SDETPLHD 可以通过调整分组次数 k 改变生成的样本点个数。为了探究分组次数对算法空间点分布带来的影响, 分组次数 k 依次设置为 1~5, 记录每种参数配置下的 SDETPLHD 算法生成点集的 Φ_p , 如表 1 所示。

表 1 分组次数运行结果

Tab. 1 Operation result of grouping times

试验空间		分组次数 k				
n_p	n_v	1	2	3	4	5
6	16	0.268 575	0.272 405	0.522 462	0.523 564	0.536 217
	25	0.222 637	0.250 030	0.522 462	0.522 462	0.524 563
	36	0.268 575	0.250 037	0.521 233	0.521 233	0.544 342
8	16	0.268 575	0.272 324	0.274 542	0.276 126	0.277 361
	25	0.218 319	0.221 367	0.223 169	0.224 457	0.225 461
	36	0.184 237	0.186 809	0.188 330	0.189 417	0.190 264
10	16	0.129 089	0.130 891	0.132 027	0.202 792	0.202 792
	25	0.131 957	0.133 799	0.134 888	0.168 993	0.168 993
	36	0.133 517	0.135 279	0.136 345	0.166 667	0.166 667
12	16	0.103 271	0.104 713	0.105 565	0.106 175	0.106649
	25	0.105 565	0.107 039	0.107 911	0.185 330	0.109 019
	36	0.106 649	0.108 138	0.109 019	0.109 643	0.110 138
14	16	0.085 185	0.086 374	0.087 077	0.087 580	0.087 971
	25	0.087 860	0.089 076	0.089 801	0.090 379	0.090 724
	36	0.088 648	0.089 885	0.090 617	0.091 140	0.094 548

实验结果表明随着分组次数 k 增加, SDETPLHD 生成点集的空间填充性变差, 但增量减小。因此, 当仿真实验需要较多采样点时, 可以增加分组次数, 此时仍能保持在较小的变化范围内, 设计点集的空间填充性较优。

4.1.2 3 种 LHD 方法对比

将 SDETPLHD、ETPLHD、lhsDesign 进行算法效率和空间填充性的对比, 结果如表 2 所示。实验结果表明在中高维度、小中大型水平数的试验空间中, SDETPLHD 的空间填充性明显比

ETPLHD 更优, 与 lhsDesign 性能相近; 在这 3 种算法中, SDETPLHD 的运行效率最高, 运行时间最短。

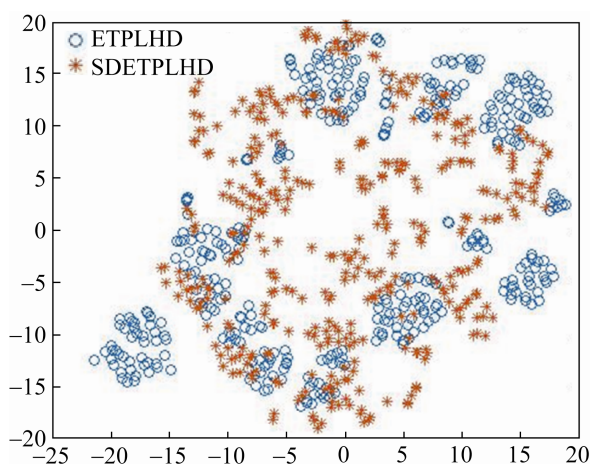
4.2 设计点集降维可视化

在中高维、小型水平配置下, SDETPLHD 与 ETPLHD 算法都生成 400 个设计点, 通过 t-SNE 算法进行降维。图 4 表明在中高维空间, SDETPLHD 算法比 ETPLHD 算法有更优的空间分布均匀性。

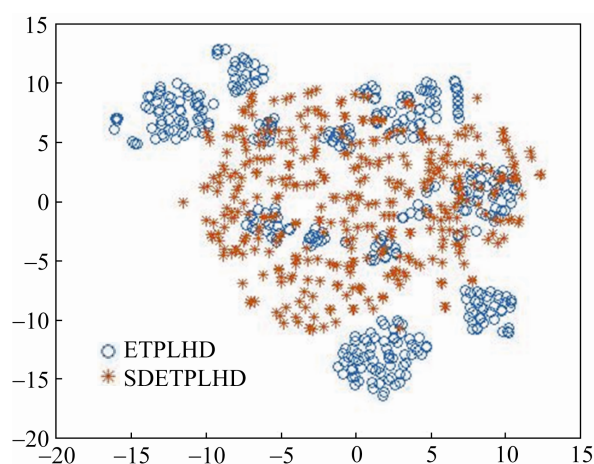
表 2 3 种算法性能对比

Tab. 2 Performance comparison of three algorithms

试验空间		SDETPLHD		ETPLHD		lhsdesign	
np	nv	faip	Time/s	faip	Time/s	faip	Time/s
6	16	0.268 6	0.028	0.268 6	0.023	0.244 4	0.337
	25	0.222 6	0.117	0.222 6	0.099	0.207 6	0.912
	36	0.223 5	0.166	0.223 5	0.135	0.183 2	1.039
8	16	0.268 6	0.113	0.345 3	0.246	0.115 2	0.511
	25	0.218 3	0.117	0.462 2	0.427	0.192 8	1.029
	36	0.184 2	0.170	0.412 1	0.533	0.176 0	1.268
10	16	0.129 1	0.036	0.292 6	0.362	0.141 2	1.016
	25	0.132 0	0.038	0.312 4	1.825	0.137 0	1.469
	36	0.133 5	1.127	0.215 4	3.111	0.126 7	2.125
12	16	0.103 3	0.039	0.310 2	1.326	0.093 1	2.435
	25	0.105 6	0.038	0.347 7	4.581	0.094 7	3.015
	36	0.106 6	1.18	0.265 4	12.39	0.086 9	3.408
14	16	0.085 2	0.125	0.210 4	7.885	0.062 5	4.256
	25	0.087 9	0.120	0.177 9	12.530	0.054 4	5.271
	36	0.088 7	1.536	0.200 2	22.150	0.050 1	5.610



(a) 8 因子×16 水平空间



(b) 10 因子×16 水平空间

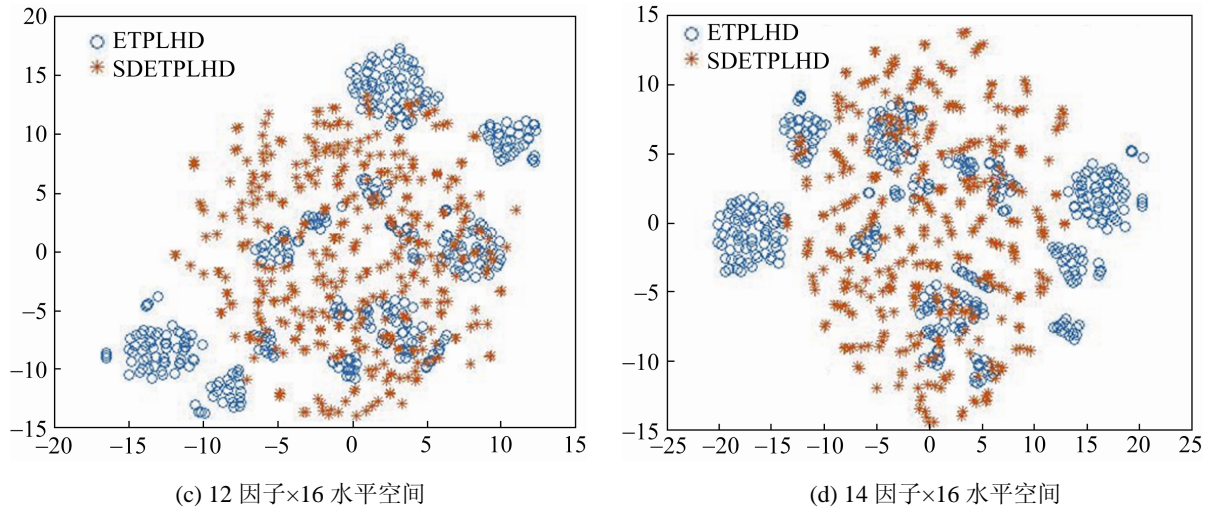


图 4 ETPLHD 与 SDETPPLHD 算法对比可视化

Fig. 4 Visualization contrast display of EDETPPLHD and SDETPPLHD

5 结论

本文提出了一种基于数独分组的扩展平移拉丁超立方设计算法 SDETPPLHD, 该算法利用数独在行列分布上的均匀性, 通过数独分组降维, 降低在中高维度的计算复杂性。在 6~14 维、16~36 水平的试验空间中, 通过与其他 2 种算法对比实验, 验证了在中高维的试验空间中, 本文设计的 SDETPPLHD 算法有更好的空间填充性和运行时间上的优越性, 可以为复杂多因子仿真提供快速较优的试验设计。

参考文献:

- [1] 别晓峰, 董艳, 王剑飞. 作战仿真实验设计的基本问题探讨[J]. 军事运筹与系统工程, 2008, 22(4): 51-56.
Bie Xiaofeng, Dong Yan, Wang Jianfei. Discussion on the Basic Problems of Combat Simulation Experiment Design[J]. Military Operation Research and System Engineering, 2008, 22(4): 51-56.
- [2] McKay M D, Beckman R J, Conover W J. A Comparison of Three Methods for Selecting Values of Input Variables from a Computer code[J]. Technometrics (S1537-2723), 1979, 42(1): 239-245.
- [3] Iman R L, Conover W J. Small Sample Sensitivity Analysis Techniques for Computer Models, with an Application to Risk Assessment[J]. Communications in Statistics Part A: Theory and Methods (S0361-0926), 1980, 9(17): 1749-1842.
- [4] Xing D D, Zhu M Y. Simulation Screening Experiments Using Lasso-optimal Supersaturated Design and Analysis: a Maritime Operations Application[C]. The 2013 Winter Simulation Conference. Washington D.C, IEEE Press, 2013: 497-508.
- [5] Jin R, Chen W, Sudjianto A. An Efficient Algorithm for Constructing Optimal Design of Computer Experiments[J]. Journal of Statistical Planning and Inference (S0378-3758), 2005, 134(1): 268-287.
- [6] Bates S J, Sienz J, Toropov V V. Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm[C]. Forty-Fifth AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. Palm Springs, CA, 2004: 19-22.
- [7] Iliedvendahl M, Stocki R. A Study on Algorithms for Optimization of Latin Hypercubes[J]. Journal of Statistical Planning and Inference (S0378-3758), 2006, 136(9): 3231-3247.
- [8] Grosso A, Jamali A, Locatelli M. Finding Maximin Latin Hypercube Designs by Iterated Local Search Heuristics[J]. European Journal of Operational Research (S0377-2217), 2009, 197(2): 541-547.
- [9] Felipe A C Viana, Gerhard Venter, Vladimir Balabanov. An Algorithm for Fast Optimal Latin Hypercube Design of Experiments[J]. International Journal for Numerical Methods in Engineering (S1097-0207), 2009, 82(2): 135-156.

- [10] Zhai Gang, Ma Yaofei, Song Xiao, et al. A Novel Experiment Design Method[C]. 2015, Asia Simulation Conference. Singapore Springer, 2015: 28-39.
- [11] Morris M D, Mitchell T J. Exploratory Designs for Computational Experiments[J]. Journal of Statistical Planning and Inference (S0378-3758), 1995, 43(3): 381-402.
- [12] Maaten L V D, Hinton G. Visualizing Data Using t-SNE[J]. Journal of Machine Learning Research (S1532-4435), 2008, 9(1): 2579-2605.
- [13] Maaten L V D. Accelerating t-SNE using Tree-Based Algorithms[J]. Journal of Machine Learning Research (S1532-4435), 2014, 15(1): 3221-3245.