Biometrika (2010), **97**, 4, *pp.* 961–968
doi: 10.1093/biomet/asq051
Advance Access publication 13 October 2010

Miscellanea

Probability-based Latin hypercube designs for slid-rectangular regions

BY YING HUNG

Department of Statistics and Biostatistics, Rutgers, The State University of New Jersey, Piscataway, New Jersey 08854, U.S.A.

yhung@stat.rutgers.edu

YASUO AMEMIYA

Statistical Analysis and Forecasting, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598, U.S.A.

yasuo@us.ibm.com

AND CHIEN-FU JEFF WU

M. Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, U.S.A.

jeffwu@isye.gatech.edu

SUMMARY

Existing space-filling designs are based on the assumption that the experimental region is rectangular, while in practice this assumption can be violated. Motivated by a data centre thermal management study, a class of probability-based Latin hypercube designs is proposed to accommodate a specific type of irregular region. A heuristic algorithm is proposed to search efficiently for optimal designs. Unbiased estimators are proposed, their variances are given and their performances are compared empirically. The proposed method is applied to obtain an optimal sensor placement plan to monitor and study the thermal distribution in a data centre.

Some key words: Computer experiment; Heuristic search; Latin hypercube design; Sampling; Space-filling design.

1. INTRODUCTION

A data centre is an integrated facility housing multiple-unit servers, providing application services or management for data processing (Schmidt et al., 2005). A challenging design issue arises in the study of data centre thermal management. To monitor and study the thermal distribution, temperature sensors are attached to surfaces of facilities. An important question is how to allocate the sensors uniformly over the data centre to facilitate a more accurate fitting of a thermal model. An obvious approach, to use existing space-filling designs, such as Latin hypercube designs (McKay et al., 1979), has limitations because the experimental region in this case is not rectangular while most such designs are constructed for rectangular regions.

In this paper, we propose a new class of space-filling designs for experiments with a specific type of irregular region, where the desirable range of one factor depends on the level of another factor. For example, the range of the x-axis in Fig. 1 depends on the level of the y-axis. Such a region is called a slid-rectangular region. The new designs are motivated by the three-dimensional sensor location problem

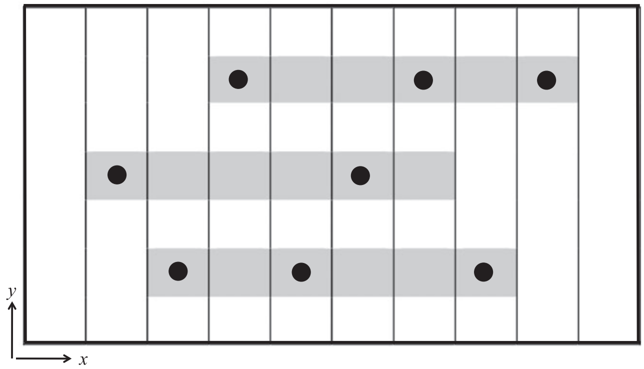

Fig. 1. An example of the probability-based Latin hypercube design. The shaded area indicates the slid-rectangular region.

but not limited to low-dimensional problems. They can be used in designing high-dimensional experiments, such as computer experiments (Santner et al., 2003; Fang et al., 2006), when the experimental region is slid rectangular.

2. PROBABILITY-BASED LATIN HYPERCUBE DESIGNS

If the n levels of a factor are denoted by $1, \ldots, n$, an n-run Latin hypercube design can be generated using a random permutation of $\{1, \ldots, n\}$ for each factor. When the experimental region is slid rectangular, a direct application of Latin hypercube designs can lose desirable space-filling properties such as one-dimensional balance. We propose a new class of designs, named probability-based Latin hypercube designs, to tackle the foregoing problem. The basic idea is to take into account the irregularity in the design construction, so that the design maintains one-dimensional balance. To illustrate the construction of such a design, consider the region in Fig. 1 but assume that eight points are to be placed. Denote them by $u_1, \ldots, u_8$, where $u_i = (x_i, y_i)$. A desirable design should have one-dimensional balance on the x-axis. Thus, one has to first divide the x-axis into eight mutually exclusive intervals. To maintain the one-dimensional balance on the x-axis, x_i should equal i, for all $i = 1, \ldots, 8$. For choosing the y values in u_i, the first step is to define the corresponding feasible range, C_i, on the y-axis for each level of x. In Fig. 1, $C_1 = \{2\}$, $C_2 = \{1, 2\}$, $C_3 = \{1, 2, 3\}$, etc. Next, for each level of x, assign the corresponding feasible y_i with equal probability. For example, $\text{pr}(y_2 = 1) = \text{pr}(y_2 = 2) = 1/2$ and $\text{pr}(y_2 = 3) = 0$ because $y_2 = 3$ is not included in the feasible set C_2. In general, this probability can be written as $\text{pr}(y_i = j) = \{\sum_{k=1}^{3} I(k \in C_i)\}^{-1}$ if $j \in C_i$; and 0, otherwise, for $i = 1, \ldots, 8$ and $j = 1, 2, 3$. The indicator function $I(k \in C_i)$ takes value 1 for $k \in C_i$ and 0 otherwise. Figure 1 shows a probability-based Latin hypercube design generated by this procedure.

A general construction procedure for probability-based Latin hypercube designs is given as follows. Hereafter, assume there are p factors and the first two factors, x_1 and x_2, form a slid-rectangular region. Here, x_2 can be a quantitative or qualitative factor with k predetermined levels and the ranges for x_1 are located irregularly on the interval $[A, B]$. Specifically, for the jth level of x_2, the feasible interval for x_1 is denoted by $E_j = (A_j, B_j)$. Thus, we have $A = \min\{A_j\}$ and $B = \max\{B_j\}$ for $j = 1, \ldots, k$. The remaining factors satisfy the standard assumptions that they are quantitative and independent. Assume that we plan for an n-run experiment and each point u_i can be represented by $u_i = (x_{1i}, \ldots, x_{pi})'$, where $i = 1, \ldots, n$ and $n \geqslant k$. According to the procedure, one-dimensional balance on x_1 is achieved by dividing the interval $[A, B]$ into n equally spaced subintervals and assigning the n levels of x_1 at the middle of these subintervals, i.e. $x_{11} = 1, \ldots, x_{1n} = n$. Then, for each level of x_1, the feasible range for x_2 is defined by C_i, and the level of x_2 is assigned by

$$\text{pr}(x_{2i} = j) = \begin{cases} \left\{\sum_{j=1}^{k} I(j \in C_i)\right\}^{-1}, & j \in C_i, \\ 0, & \text{otherwise,} \end{cases} \quad (j = 1, \ldots, k, \quad i = 1, \ldots, n). \tag{1}$$

For the last $p - 2$ variables, standard construction procedures for Latin hypercube designs can be applied.

Using standard results in probability sampling (Cochran, 1977), an unbiased estimator of the population mean based on the probability-based Latin hypercube designs can be given as

$$T = N^{-1} \sum_{i=1}^{n} \sum_{j \in C_i} E(w_{ij})^{-1} w_{ij} g(Y_{ij}), \tag{2}$$

where $g(\cdot)$ is an arbitrary function, Y_{ij}s are the responses, w_{ij} is an indicator variable with $w_{ij} = 1$ if Y_{ij} is selected by the design, and $w_{ij} = 0$ otherwise. It is clear that $E(w_{ij}) = c_i^{-1}$, where $c_i = \sum_{l=1}^{k} I(l \in C_i)$. In survey sampling, T is the Horvitz–Thompson estimator. Its variance can be written in the Yates–Grundy expression (Cochran, 1977, pp. 260–1):

$$\text{var}(T) = \tfrac{1}{2} N^{-2} \left[\sum_{ij} \sum_{qt(qt \neq ij)} (\pi_{ij}\pi_{qt} - \pi_{ij,qt}) \left\{ \frac{g(Y_{ij})}{\pi_{ij}} - \frac{g(Y_{qt})}{\pi_{qt}} \right\}^2 \right], \tag{3}$$

with $\pi_{ij} = E(w_{ij}) = c_i^{-1}$, $\pi_{ij,it} = 0$ for $t \neq j$, and $\pi_{ij,qt} = \pi_{ij}\pi_{qt} = c_i^{-1}c_q^{-1}$ for $i \neq q$, where $\pi_{ij,qt}$ is the joint selection probability of unit (i, j) and unit (q, t). An unbiased variance estimator can be readily obtained by replacing N in (3) by n and the population values Y_{ij} in (3) by the corresponding sample values.

For a given number of runs and factors, probability-based Latin hypercube designs are not unique. Considering all the possible designs, Proposition 1 gives the average number of points with level j of factor x_2. The proof is straightforward and omitted.

PROPOSITION 1. *Let n_j denote the number of points with $x_2 = j$. Then*

$$E(n_j) = \sum_{i=1}^{n} \text{pr}(x_{2i} = j) = \sum_{i=1}^{n} \frac{I(j \in C_i)}{\sum_{j=1}^{k} I(j \in C_i)}.$$

Take the probability-based Latin hypercube design in Fig. 1 as an example. Proposition 1 gives the expected number of points located in each shaded area, i.e. $4/3 + 1/2 + 1 = 17/6$ for the upper shaded area, $1 + 1/2 + 4/3 = 17/6$ for the middle one and $1/2 + 4/3 + 1/2 = 17/3$ for the lower one.

3. BALANCED PROBABILITY-BASED LATIN HYPERCUBE DESIGNS

Ideally, the n_j values in Proposition 1 should be proportional to the length of the shaded area, which can be written as $B_j - A_j = \sum_{i=1}^{n} I(j \in C_i)$ and $j = 1, \ldots, k$. This is because the information from each area is assumed to be proportional to its length. According to Proposition 1, this is not necessarily true unless the values, $\sum_{j=1}^{k} I(j \in C_i)$, are the same for all i. Inspired by this observation, a further modification is introduced to incorporate the proportional balance property, where the number of observations is proportional to the length of the interval. We call it balanced probability-based Latin hypercube design, which can be written as a modification of (1) with the constraints

$$\frac{n_j}{n} = \frac{(B_j - A_j)}{\sum_{j=1}^{k}(B_j - A_j)} = p_j \qquad (j = 1, \ldots, k). \tag{4}$$

In practice, the quantities np_j are not always integers for given n and p_j. In that situation, an approximate balance with $|n_j - np_j| < 1$ should be imposed.

Balanced probability-based Latin hypercube designs are constrained probability-based Latin hypercube designs in which the proportional allocation property is maintained. Table 1 lists such a design with three factors $p = 3$ and 22 runs. For the slid-rectangular region, factor x_2 has five levels and the proportional lengths of x_1 at different levels of x_2 are 3:4:5:6:4. Both the proportional allocation property and the one-dimensional balance property hold for the first two factors.

There are different ways to construct balanced probability-based Latin hypercube designs, and the unbiased estimator T in (2) is defined according to their sampling probabilities. An unbiased estimator

Table 1. *An example of a balanced probability-based Latin hypercube design*

x_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
x_2	1	1	2	2	1	3	4	4	2	3	4	5	5	2	3	4	5	3	3	4	5	4
x_3	18	1	8	14	20	4	11	17	7	21	3	15	10	19	5	13	22	9	16	2	12	6

is given below with the following drawing rule. The units are sequentially selected based on (1) from $i = 1$ to n. For each j selected by the algorithm, a design point is assigned to the unit only if (4) is not yet achieved. If it is already achieved, discard the current design and restart the selection from $i = 1$. The algorithm continues until $i = n$ is reached and a design point is assigned. This drawing rule is called rejection sampling in the simulation literature (Robert & Casella, 2005). Based on the above drawing rule, the sampling probability $E(w_{ij})$ can be written as

$$Q_{ij} = \left[c_i^{-1}\mathrm{pr}(a_{ij} = 1 \mid 1, \ldots, i-1)\prod_{l=2}^{i-1} c_l^{-1}\left\{\sum_m \mathrm{pr}(a_{lm} = 1 \mid 1, \ldots, l-1)\right\}\right.$$
$$\left.\times \prod_{l>i} c_l^{-1}\left\{\sum_m \frac{\mathrm{pr}(a_{lm} = 1, a_{ij} = 1 \mid 1, \ldots, i-1, i+1, \ldots, l-1)}{\mathrm{pr}(a_{ij} = 1 \mid 1, \ldots, i-1, i+1, \ldots, l-1)}\right\}\right]$$
$$\times \left[1 - \sum_{h=2}^{n} c_h^{-1}\left\{\sum_m \mathrm{pr}(a_{hm} = 0 \mid 1, \ldots, h-1)\right\}\prod_{l=2}^{h} c_{l-1}^{-1}\left\{\sum_m \mathrm{pr}(a_{l-1,m} = 1 \mid 1, \ldots, l-2)\right\}\right]^{-1}, \tag{5}$$

where a_{ij} is an indicator variable with $a_{ij} = 1$ if a design point is successfully assigned to cell (i, j) at that stage, and $a_{ij} = 0$ otherwise. The function $\mathrm{pr}(a_{ij} = 1 \mid 1, \ldots, i-1)$ is the probability of assigning design points to cell (i, j) provided that the first $i-1$ points have been successfully assigned. As in § 2, $T_\mathrm{B} = N^{-1}\sum_i \sum_j Q_{ij}^{-1} w_{ij} g(Y_{ij})$ is an unbiased estimator of the population mean based on a balanced probability-based Latin hypercube design if $Q_{ij} > 0$ for all i and j. The variance of T_B can be written in the form of (3) with $\pi_{ij} = Q_{ij}$, $\pi_{ij,it} = 0$, and

$$\pi_{ij,qt} = Q_{ij}\frac{\mathrm{pr}(a_{qt} = 1, a_{ij} = 1 \mid 1, \ldots, i-1, i+1, \ldots, n)}{\mathrm{pr}(a_{ij} = 1 \mid 1, \ldots, i-1, i+1, \ldots, n)} \quad (t \geqslant j, \quad q > i). \tag{6}$$

Proofs of (5) and (6) are given in the Appendix. For any given slid-rectangular region, the values of Q_{ij} can be calculated explicitly. If $Q_{ij} = 0$ for one pair of i and j, it would be undesirable to use the balanced design because it would completely miss the Y_{ij} value in its sampling. This does not appear to occur often. When it occurs, the choice should be reverted to the original probability-based Latin hypercube designs whose $E(w_{ij})$ values are always positive.

Simulations are conducted to compare T_B with T and another unbiased estimator from a simple random sample of the same size, denoted by T_SRS. The responses Y, as a function of t, are generated from a normal distribution with variance 1 and mean function, $\beta_1 x_1 + \beta_2 x_2 + \cos(0{\cdot}5\pi t)$. The factors x_1 and x_2 form a slid-rectangular region as shown in Table 1, where the values of Q_{ij} are positive. The mean function is designed to decrease with t, and β_1 and β_2 are used to control the effect of experimental region on the mean function. Comparisons are based on $g\{Y(t)\} = Y(t)$ with 100 t values uniformly selected from (0, 1) and two settings of the β_is, $\beta_1 = \beta_2 = 0{\cdot}01$ and $0{\cdot}1$. The means and standard deviations of the three estimators are computed based on 50 simulations for each t. For both settings of β_i, the means of the three estimators are comparable, demonstrating the unbiasedness of these estimators. For the standard deviations, the three estimators perform similarly for small β_is. For large β_is, T_B has smaller standard deviations than T, and T_SRS has larger standard deviations. This observation has an analogy in survey sampling (Cochran, 1977), i.e. the estimation variance for simple random sampling is at least as large as that for stratified random sampling, which is in turn at least as large as that for stratified random sampling with proportional allocation.

4. A Search Algorithm

Since not all probability-based Latin hypercube designs or balanced probability-based Latin hypercube designs are equally good, some optimal design criteria are needed for further discrimination. The criteria used for Latin hypercube designs (Iman & Conover, 1982; Johnson et al., 1990; Owen, 1994; Morris & Mitchell, 1995; Tang, 1998) can be easily applied to the designs considered here. For example, the design shown in Table 1 is an optimal balanced probability-based Latin hypercube design according to the maximin criterion (Johnson et al., 1990), where the objective is to maximize the minimum intersite distance.

A design is called feasible if the points lie in the slid-rectangular region. The usual search algorithm for optimal Latin hypercube designs cannot be directly applied here because arbitrary exchange of two elements within a column does not always lead to a feasible design. A naive approach is to modify the algorithm by allowing only random exchanges that are capable of producing a feasible solution. Using this procedure, a sequence of feasible designs is generated and examined by the optimality criterion. Because of the combinatorial nature of the problem, finding optimal probability-based Latin hypercube designs can be difficult. A complete search is computationally prohibitive for large problems. Therefore an efficient heuristic algorithm is needed.

To improve the efficiency of the naive search, we introduce a new algorithm. The idea is to broaden the search by preventing visits of neighbourhood designs that have been visited before. Neighbourhood designs are those that differ from each other by a small number of columnwise-pairwise exchanges. Because the neighbourhood designs are similar, avoiding them would allow the search to move to other parts of the region with more promising values. To do so, the naive search is modified by keeping track of the previous q feasible settings of x_2 visited, which are called forbidden settings, with q being a tuning parameter. The forbidden setting is also called the memory in the tabu search literature (Glover, 1986). Since a visit to the forbidden settings would lead to a movement towards the neighbourhood designs, identification of the forbidden settings/memory can be effectively used to prevent the current design from moving towards neighbourhoods. This makes the search more efficient. In fact, the forbidden settings can be defined based on any factor. However, a good choice is x_2 because the verification of feasible exchanges in x_2 is more time-consuming. If the x_2 candidates lie in the forbidden set, they are removed from consideration immediately without verifying their feasibility. Thus, some computation can be saved and the x_2 settings are explored more efficiently. Typically the tuning parameter q should be small in order to maintain a small neighbourhood. This algorithm is called a columnwise-pairwise exchange tabu algorithm.

After specifying a design criterion, the proposed algorithm begins with a randomly chosen design X, and proceeds with the examination of a sequence of designs. Each design is generated as follows. First, a column from x_2 to x_p is randomly selected. If the selected column is one of the last $p-2$ factors, a new design is obtained by exchanging two randomly selected elements within the column. If the column of x_2 is selected, it has to be checked whether the exchange of two randomly selected x_2 levels leads to a setting in the forbidden set, i.e., if it is among the last q feasible settings visited. If not, the feasibility has to be checked, i.e., whether the resulting design lies in the slid-rectangular region. The exchange is allowed only if the resulting new setting is non-forbidden and feasible. Otherwise, random exchanges within the x_2 column continue to be examined until a new feasible setting is obtained. Following this procedure, new designs X^{try} are generated. In each iteration, X is replaced by X^{try} if it leads to an improvement with respect to the design criterion. Once values of the design criterion are stabilized, the algorithm is terminated and the resulting X is the optimal design. This algorithm also works for balanced probability-based Latin hypercube designs with carefully constructed initial designs because the proposed procedure maintains the balance property throughout the search if the initial design is balanced.

To assess its performance, the proposed algorithm with $q = 7$ is applied to the search for the optimal maximin balanced probability-based Latin hypercube design listed in Table 1. With the use of the forbidden settings, the simulation results show that the new algorithm can on average obtain designs with scores 4·1% smaller than the naive method after 1500 iterations.

5. Data centre example

The proposed method is applied to a sensor placement problem in a data centre located in the IBM T. J. Watson Research Center. The two-dimensional layout of the data centre is illustrated in Fig. 2. Four racks

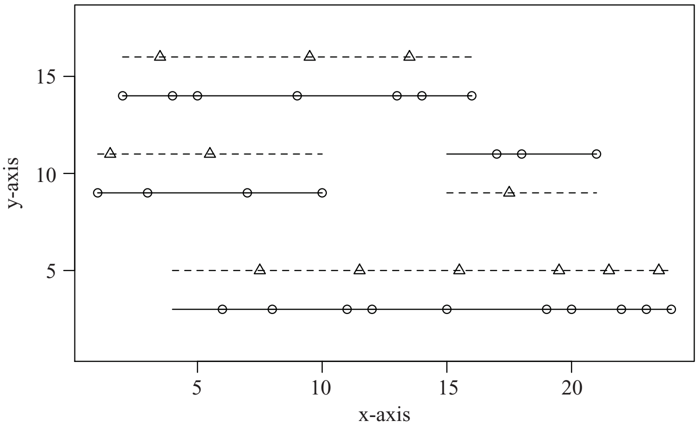

Fig. 2. Sensor locations in an actual data centre. Four racks are located in three rows. Inlet and outlet surfaces are denoted by solid and dashed lines, respectively. Circles are the inlet sensors and triangles are the outlet sensors.

with different lengths are located in three rows. Each rack includes an inlet surface and an outlet surface. The layout of the racks result in a slid-rectangular region that can be specified by the first two coordinates, x and y, and the sensor locations are represented by the three-dimensional coordinates, x, y and z, where the z-axis represents the height.

In this experiment 36 sensors are available. As monitoring and controlling the inlet temperature is more important than the outlet temperature, more sensors should be placed on the inlet surfaces than the outlet surfaces, so 24 sensors are assigned to the inlet surfaces and 12 to the outlet surfaces. According to Fig. 2, four levels on the y-axis are predetermined based on the location of the inlet surfaces. Hence, we have $k = 4$ and $n = 24$. A 24-run maximin balanced probability-based Latin hypercube design is generated for the inlet sensors. Similarly, design points for the outlet surfaces are generated with $k = 4$ and $n = 12$. Since fewer points are located on the outlet surface, the proportional allocation property is relatively difficult to maintain. The sensor placement plan for the outlet surface should employ a 12-run maximin probability-based Latin hypercube design. Optimal designs for inlet and outlet surfaces are generated using the columnwise-pairwise exchange tabu algorithm with $q = 7$ and $q = 5$, respectively.

ACKNOWLEDGEMENT

The authors are grateful to a reviewer for pointing out an error leading to a substantial improvement of the paper and to another reviewer for helpful editorial comments. Their research is supported by grants from the National Science Foundation, U.S.A.

APPENDIX

Proof of (5). Recall that $w_{ij} = 1$ if cell (i, j) belongs to a balanced probability-based Latin hypercube design. It follows that

$$\begin{aligned}\text{pr}(w_{ij} = 1) = {} & \text{pr}(w_{ij} = 1 \text{ in the first round}) \\ & + \sum_{r=1}^{\infty} \text{pr}(\text{the first } r \text{ rounds are discarded and } w_{ij} = 1 \text{ in the } r + 1\text{round}) \\ = {} & \text{pr}(w_{ij} = 1 \text{ in the first round}) \\ & + \sum_{r=1}^{\infty} \text{pr}(\text{one round is discarded})^r \, \text{pr}(w_{ij} = 1 \text{in the first round}).\end{aligned} \quad \text{(A1)}$$

Define $s_{ij} = 1$ if cell (i, j) is selected by (1) and $d_{ij} = 1$ if each level in x_1, except the ith level, has one design point assigned. Recall that $a_{ij} = 1$ if a design point can be successfully assigned to cell (i, j) at stage i. Hence, we have

$$
\begin{aligned}
\text{pr}(w_{ij} = 1 \text{in the first round}) &= \text{pr}(s_{ij} = 1)\text{pr}(a_{ij} = 1, d_{ij} = 1) \\
&= c_i^{-1}\text{pr}(a_{ij} = 1, d_{ij} = 1) \\
&= c_i^{-1}\text{pr}(D_1)\left\{\prod_{l=2}^{i-1}\text{pr}(D_l \mid D_1, \ldots, D_{l-1})\right\}\text{pr}(a_{ij} = 1 \mid D_1, \ldots, D_{i-1}) \\
&\quad \times \left\{\prod_{l>i}^{n}\text{pr}(D_l \mid D_1, \ldots, D_{i-1}, a_{ij} = 1, D_{i+1}, \ldots, D_{l-1})\right\}, \qquad \text{(A2)}
\end{aligned}
$$

where D_i is the event that the ith level of x_1 has one design point assigned, i.e. $D_i = \left(\sum_m a_{im} = 1\right)$. It is clear that $P(D_1) = P\left(\sum_m a_{1m} = 1\right) = 1$ and for $l < i$, we have

$$
\text{pr}(D_l \mid D_1, \ldots, D_{l-1}) = c_l^{-1}\left\{\sum_m \text{pr}(a_{lm} = 1 \mid D_1, \ldots, D_{l-1})\right\}. \qquad \text{(A3)}
$$

Similarly, for $l > i$, we have

$$
\begin{aligned}
&\text{pr}(D_l \mid D_1, \ldots, D_{i-1}, a_{ij} = 1, D_{i+1}, \ldots, D_{l-1}) \\
&= c_l^{-1}\left\{\sum_m \text{pr}(a_{lm} = 1 \mid D_1, \ldots, D_{i-1}, a_{ij} = 1, D_{i+1}, \ldots, D_{l-1})\right\} \\
&= c_l^{-1}\left\{\sum_m \frac{\text{pr}(a_{lm} = 1, a_{ij} = 1 \mid D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_{l-1})}{\text{pr}(a_{ij} = 1 \mid D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_{l-1})}\right\}. \qquad \text{(A4)}
\end{aligned}
$$

For the probability that one round is discarded, we have

$$
\begin{aligned}
&\text{pr}(\text{one round is discarded}) \\
&= \sum_{h=2}^{n}\text{pr}(\text{one round is discarded at stage } h) \\
&= \sum_{h=2}^{n}\text{pr}\left(D_h^c \mid D_1, \ldots, D_{h-1}\right)\left\{\prod_{l=2}^{h}\text{pr}(D_{l-1} \mid D_1, \ldots, D_{l-2})\right\} \\
&= \sum_{h=2}^{n} c_h^{-1}\left\{\sum_m \text{pr}(a_{hm} = 0 \mid D_1, \ldots, D_{h-1})\right\}\left[\prod_{l=2}^{h} c_{l-1}^{-1}\left\{\sum_m \text{pr}(a_{l-1,m} = 1 \mid D_1, \ldots, D_{l-2})\right\}\right]. \qquad \text{(A5)}
\end{aligned}
$$

From (A1)–(A5), $E(w_{ij}) = Q_{ij}$ holds with the simplified notation $\text{pr}(a_{ij} = 1 \mid 1, \ldots, i-1)$ instead of $\text{pr}(a_{ij} = 1 \mid D_1, \ldots, D_{i-1})$. □

Proof of (6). The joint selection probability $\pi_{ij,qj}$ can be written as

$$
\begin{aligned}
\text{pr}(w_{qj} = 1, w_{ij} = 1) &= \text{pr}(w_{qj} = 1, w_{ij} = 1 \text{ in the first round})\,\{1 - \text{pr}(\text{one round is discarded})\}^{-1} \\
&= \text{pr}(w_{qj} = 1 \text{ in the first round} \mid w_{ij} = 1 \text{ in the first round})\text{pr}(w_{ij} = 1).
\end{aligned}
$$

Without loss of generality, assume $q > i$ and $t \geqslant j$. Then, we have

$$
\pi_{ij,qt} = \text{pr}(w_{qt} = 1 \text{ in the first round} \mid w_{ij} = 1 \text{ in the first round})\text{pr}(w_{ij} = 1),
$$

where

$$\begin{aligned}\text{pr}(w_{qt} &= 1\text{in the first round } \mid w_{ij} = 1 \text{ in the first round})\\ &= \text{pr}(a_{qt} = 1 \mid w_{ij} = 1 \text{ in the first round})\text{pr}(d_{qt} = 1 \mid w_{ij} = 1, a_{qt} = 1 \text{ in the first round})\\ &= \text{pr}(a_{qt} = 1 \mid D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n, a_{ij} = 1)\\ &= \frac{\text{pr}(a_{qt} = 1, a_{ij} = 1 \mid D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n)}{\text{pr}(a_{ij} = 1 \mid D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n)}.\end{aligned}$$

Therefore (6) follows. □

REFERENCES

COCHRAN, W. G. (1977). *Sampling Techniques*, 3rd ed. New York: Wiley.

FANG, K. T., LI, R. & SUDJIANTO, A. (2006). *Design and Modeling for Computer Experiments*. New York: CRC Press.

GLOVER F. (1986). Future paths for integer programming and links to artificial intelligence. *Comp. Oper. Res.* **13**, 533–49.

IMAN, R. L. & CONOVER, W. J. (1982). A distribution-free approach to inducing rank correlation among input variables. *Commun. Statist. B* **11**, 311–34.

JOHNSON, M., MOORE, L. & YLVISAKER, D. (1990). Minimax and maximin distance design. *J. Statist. Plan. Infer.* **26**, 131–48.

MCKAY, M. D., BECKMAN, R. J. & CONOVER, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–45.

MORRIS, M. D. & MITCHELL, T. J. (1995). Exploratory designs for computer experiments. *J. Statist. Plan. Infer.* **43**, 381–402.

OWEN, A. (1994). Controlling correlations in Latin hypercube samples. *J. Am. Statist. Assoc.* **89**, 1517–22.

ROBERT, C. P. & CASELLA, G. (2005). *Monte Carlo Statistical Methods*, 2nd ed. New York: Springer.

SANTNER, T., WILLIAMS, B. & NOTZ, W. (2003). *The Design and Analysis of Computer Experiments*. New York: Springer.

SCHMIDT, R. R., CRUZ, E. E., & IYENGAR, M. K. (2005). Challenges of data center thermal management. *IBM. J. Res. Dev.* **49**, 709–23.

TANG, B. (1998). Selecting Latin hypercubes using correlation criteria. *Statist. Sinica.* **8**, 965–78.

[*Received August* 2009. *Revised May* 2010]