Using integer programming techniques for the solution of an experimental design problem

Carl M. Harris\(^1\) and Karla L. Hoffman\(^2\)

Department of Operations Research and Engineering,
George Mason University, Fairfax, VA 22030, USA

Leslie-Ann Yarrow\(^3\)

Chesapeake Decision Sciences, Inc.,
New Providence, NJ 07974, USA

Latin hypercube sampling is often used to estimate the distribution function of a complicated function of many random variables. In so doing, it is typically necessary to choose a permutation matrix which minimizes the correlation among the cells in the hypercube layout. This problem can be formulated as a generalized, multi-dimensional assignment problem. For the two-dimensional case, we provide a polynomial algorithm. For higher dimensions, we offer effective heuristic and bounding procedures.

**Keywords:** Assignment problem, computer models, distribution sampling, estimation, integer programming, large-scale modelling, latin hypercube, optimization, sampling, sensitivity analysis.

1. Introduction

The focus of this work is to consider the solution of a sampling design problem using combinatorial optimization. The particular design problem of interest here is minimum-correlation latin hypercube sampling (hereafter referred to as MCLHS). This problem is important to those interested in experiments where the sampling is extremely expensive. The problem is also interesting from the integer programming perspective. This problem seems “easy” in that the duality gap between the optimal integer programming solution value and the optimal linear programming relaxation

---

\(^1\)Supported in part by a grant from the National Institute of Standards and Technology (60NANB9D-0974).
\(^2\)Supported in part by grants from the Office of Naval Research (N00014-90-J-1324) and the Air Force Office of Scientific Research (F49 620-90-C-0022).
\(^3\)Research partially performed while visiting the Department of Mathematics, Brunel University, Uxbridge, England.
value is zero, but there are a plethora of fractional solutions and one must hunt for an integer solution among these fractional vertices.

The primary objective of Latin hypercube sampling (LHS) is to estimate the probability distribution function of a complicated function of many random variables. It can be used to empirically derive a distribution function that has otherwise been difficult to determine or for which there is no closed-form representation. The primary problem to consider is what is the most efficient way to sample from the input population, especially when sampling is extremely expensive and must therefore be limited to a predetermined (small) sample size.

Alternatively, Latin hypercube sampling has also been used in sensitivity analysis of a variety of models. Many computer-based modeling efforts require the repeated execution of a computer model to estimate the variability in the output parameters given minor perturbations in the input parameters. These models are often expensive to run, thereby limiting the total sampling (runs) possible. Harris [5] offers an overview and introduction of the use of LHS design for evaluating the sensitivity of a model’s output to changes in the input values. For examples of such applications of LHS see, for instance, Iman and Conover [9, 10], Harris [5, 6], and Chapman et al. [3]. In the final section of this paper, we describe how the method of minimum-correlation Latin hypercube sampling could be applied to such studies.

In the application of Latin hypercube sampling, there may or may not exist non-zero correlations among the random variables of the distribution being estimated. Thus, not any LHS plan should be used, but rather one that reflects or incorporates the specific correlation structure that exists between the underlying variables. By not considering these relationships, a researcher could find that a sampling plan may introduce an implicit bias to the estimation.

Previous research on Latin hypercube sampling has either not addressed this issue, or been unsuccessful in finding plans that exactly meet pre-specified correlation requirements. Even when the requirement is that there be no intercorrelation (i.e. the correlations among all variables are zero), the random methods currently in use do not obtain these zero correlations. To our knowledge, this critical issue of how closely the sampling coincides with the variable correlations was first highlighted in Iman and Conover [10]. In that work, the authors presented an approximation method that was based on generating a variety of different plans randomly and then choosing among those plans the one which best met the intended correlation. Owen [11] proposed an alternative method that offers nice asymptotic results and uses the same measure of best (matching the Spearman rank correlation) as is used in this paper. However, these authors can only provide asymptotic convergence to such plans for very large sample sizes. Since Latin hypercube sampling plans are used only when sampling size is restricted to be small, asymptotic results are unappealing and, we believe, irrelevant. Alternatively, in this paper, we provide a methodology for obtaining sampling plans which accurately reflect the true correlation matrix even when the sample size proposed is extraordinarily small. We note that all other papers on this
subject have argued that it was not possible (due to the combinatorial nature of the problem) to find such optimal plans.

To illustrate the differences among alternative LHS plans, consider a population with three random variables, \( x \), \( y \), and \( z \) and suppose the individual CDFs are divided into eight blocks. Whereas there are \( 8^3 \) possible samples, the following sample plan defines eight samples. This LHS plan defines that run 1 will require picking randomly from \( x \)'s block 2, picking randomly from \( y \)'s block 6 and from \( z \)'s block 3. Similarly, for run 4, blocks 6, 3, and 1 will be drawn from for the fourth samples of \( x \), \( y \), and \( z \), respectively. For this plan, the Spearman rank correlation coefficients are \( r_{12} = 1 - 396/504 = 0.214 \), \( r_{23} = -0.262 \), and \( r_{13} = 0.238 \). On the other hand, the following LHS plan has rank correlation coefficients all equal to zero:

<table>
<thead>
<tr>
<th>Model run</th>
<th>Variable values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x_2 ) ( y_6 ) ( z_3 )</td>
</tr>
<tr>
<td>2</td>
<td>( x_7 ) ( y_8 ) ( z_4 )</td>
</tr>
<tr>
<td>3</td>
<td>( x_1 ) ( y_4 ) ( z_2 )</td>
</tr>
<tr>
<td>4</td>
<td>( x_6 ) ( y_3 ) ( z_1 )</td>
</tr>
<tr>
<td>5</td>
<td>( x_4 ) ( y_7 ) ( z_5 )</td>
</tr>
<tr>
<td>6</td>
<td>( x_5 ) ( y_2 ) ( x_6 )</td>
</tr>
<tr>
<td>7</td>
<td>( x_3 ) ( y_1 ) ( z_8 )</td>
</tr>
<tr>
<td>8</td>
<td>( x_8 ) ( y_5 ) ( z_7 )</td>
</tr>
</tbody>
</table>

In Harris, Hoffman and Yarrow [7] (HHY), we have identified inherent characteristics of the MCLHS design problem, which were then employed in a heuristic procedure that consistently offers optimal plans, and when it does not, is probably very close to optimality. In this paper, we describe a methodology that builds on the HHY heuristic procedure by embedding it within a Lagrangian subgradient optimization approach. We report computational testing of this algorithm.
which finds the optimal solution for all cases tried. We note that throughout most of this presentation, we address the case where the correlations among all variables are zero. The modifications needed to allow for the general correlation cases are simple extensions to the approach described here and are fully discussed in HHY.

We begin by providing a brief overview of the problem characteristics and the heuristic procedure that will be used in this work. In section 2, we present an integer programming formulation and solution approach which meets the prespecified correlation structure exactly. Empirical results are presented in section 3 and conclusions follow.

Latin hypercube sampling is a special form of stratified sampling. In this stratification, the cumulative distribution functions for each of the \( n \) population variables are divided into \( m \) blocks. The intersection of these blocks makes up a hypercube having \( m^n \) cells. Since sampling is assumed to be expensive, LHS limits the sampling to only \( m \) of the \( m^n \) possible cells. Thus, an LHS plan is not really a hypercube, but is equivalent to an \( m \times n \) matrix such that each of the \( m \) rows defines one sampling cell of an \( m^n \) hypercube. Hereafter, the \( i \)th row of a sampling plan makes up what will be referred to as "run \( i \)."

The standard approach to LHS begins by writing the vector of variables as \( X = (X_1, X_2, \ldots, X_n) \) and assumes for the time being that the variables are mutually independent. The range of each \( X_i \) is then divided into \( m \) (= number of runs) ascending intervals of equal probability and a random value is drawn from each interval for each variable.

Next, the order in which the \( m \) values of each variable are to be used in each run is generated by creating a sequence of \( n \) random permutations of the integers \( m \). Finally, the required vector for the first run is formed by taking the leading number from each of the \( n \) random permutations and continue to do this similarly matching the \( i \)th elements of each permutation until all \( m \) are used.

The key property of an LHS layout is that each set of column indices is a distinct permutation of the digits 1 through \( m \). It may also turn out that each row is a unique permutation of \( n \) digits taken from \( m \), but this is not a requirement for a classical LHS plan. Indeed, since the standard approach selects its column permutations independently, it is clearly possible to have repeat indices in one or more rows. Therein lies a critical issue, ultimately a major focus of this work. It is that the calculated correlation coefficient (say \( r \)) of the indices of any pair of generated permutations (namely, the rank correlation coefficient) can turn out to be anywhere in \([-1, 1]\), although it can be shown that the expected value of \( r \) is 0, and that its variance is \( 1/(m-1) \). Since we are dealing with permutations of the integers from 1 through \( m \), it is easily shown that the correlation coefficient between any two such vectors is

\[
r = 1 - \frac{6 \sum_{p=1}^{m} D_p^2}{m(m^2 - 1)},
\]

where \( D_p \) is the difference between the \( p \)th integer elements in the vectors.
This is, of course, the well-known Spearman rank correlation coefficient. Throughout the remainder of this paper, we denote \( r_{ij} \) to be the correlation between column permutations \( i \) and \( j \). The selection of column permutations should attempt to minimize some function of the absolute values of pairwise correlations; we consider the minimization of the sum of the absolute values of the coefficients. For more on the statistical properties of LHS plans, see HHY [7] or Yarrow [12].

2. Integer programming formulation and heuristic solution approach

One obvious formulation of the MCLHS problem is as an \( n \)-index assignment problem with side knapsack equation constraints (APSEC). To begin, define

\[
x_{v_1 \ldots v_n} = \begin{cases} 
1 & \text{if } v_1 v_2 \ldots v_n \text{ is a sampled cell where the} \\
& \text{\quad } n\text{-indices on the } x\text{-variable, } v_1, v_2, \ldots, v_n, \\
& \text{\quad } \text{can each take a value from 1 to } m; \\
0 & \text{otherwise.}
\end{cases}
\]

Since \( m(m^2 - 1)r/6 = m(m^2 - 1)/6 - \sum D_v^2 \) will be integer-valued for all pairs of permutations (see HHY [7]), we further define \( d_{ij}^+, d_{ij}^- \in \mathbb{Z}^+_1 \) such that

\[
d_{ij}^+ - d_{ij}^- = m(m^2 - 1)r_{ij}/6, \quad i = 1 \ldots n, \ j > i.
\]

The MCLHS objective to minimize the sum of the absolute values of the rank correlation coefficients is thus equivalent to

\[
\min \left\{ \sum_{i=1}^{n-1} \sum_{j>i}^{n} (d_{ij}^+ + d_{ij}^-) \right\}.
\]

In order that the IP formulation fully encompass the MCLHS, it must include assignment constraints that draw a one-to-one correspondence between the positive-values \( x_{v_1 \ldots v_n} \) of a feasible solution and \( n \)-tuples of column permutations. Thus, the \( j \)th column permutation requires the \( m \) assignment constraints

\[
\sum_{v_1=1}^{m} \sum_{v_2=1}^{m} \ldots \sum_{v_k=1}^{m} x_{v_1 \ldots v_n} = 1, \quad v_j = 1 \ldots m.
\]

excluding \( \sum_{v_j=1}^{m} \)

Additional constraints are needed to enforce that \( m(m^2 - 1)r_{ij}/6 = d_{ij}^+ - d_{ij}^- \) holds for all \( i \) and \( j \), \( i < j \leq n \). These constraints are

\[
\sum_{v_1=1}^{m} \sum_{v_2=1}^{m} \ldots \sum_{v_n=1}^{m} (v_i - v_j)^2 x_{v_1 \ldots v_n} = m(m^2 - 1)/6, \quad \forall i < j \leq n.
\]
We see that this formulation requires $m^n$ $x$-variables as well as a total of $n(n-1) d_{ij}$ variables. The number of constraints is $nm + \binom{n}{2}$. Hence, although this formulation is the most straightforward, we will present other APSEC formulations which have more reasonable problem size growth.

Another disadvantage of this formulation is that it belongs to the class of NP-complete problems. This follows, since removing the equation knapsacks from the constrained set yields the multi-dimensional assignment problem that has been shown by Balas and Saltzman [1] to be NP-hard for $n \geq 3$.

Finally, we have empirically observed that the linear programming relaxation, formed by relaxing the variables' integrality restrictions, always had an optimal solution value of zero. Thus, for $m = 2 + 4l$, since any feasible integral solution will have an objective value that is bounded from below by $n - 1$, a gap can exist between the linear and integer objective function values. We also note that for all values of $m$ and $n$ examined, the LP solution obtained by using a standard simplex-based software package, CPLEX, had fractional components.

Although a number of properties related to this problem were discussed in HHY [7], a few proved essential in the development of an effective heuristic for identifying optimal or near-optimal minimum correlation LHS plans. We restate those that are also relevant to our integer programming solution approach.

We first present a lower bound on the objective function value that can be used as a stopping criterion for any approach that attempts to find optimal MCLHS plans and also is useful in providing alternative formulations to this problem.

**Lemma**

For $m \geq 2$, and a fixed $n$, the minimum sum of the absolute values

$$Z = \min \sum_{j=2}^{n} \sum_{i=1}^{j-1} |r_{ij}|$$

is bounded from below by $(\binom{n}{2}6/|[m(m^2 - 1)]$ when $m = 2 + 4l$, for some nonnegative integer $l$, and by zero otherwise.

Rather than minimizing the sum of the absolute values of $r$, we minimize the sum of the absolute values

$$|m(m^2 - 1)r/6| = |m(m^2 - 1)/6 - \sum D_{ij}^2|,$$

since an optimal solution vector to the latter objective function yields an optimal solution vector to the former. However, the latter will always yield an integral value, whereas the former can yield fractions close to zero that cause numerical difficulties in accurately determining an optimal solution. From the lemma, we have that a permutation matrix is optimal if for each column pair,
\[ |m(m^2 - 1)/6 - \sum D^2_i| = \begin{cases} 1 & \text{if } m = 2 + 4l, \ l \in \mathbb{Z}_1^+, \\ 0 & \text{otherwise}. \end{cases} \]

In HHY [7], we have empirically found two other properties that impact our ability to better formulate this problem. The first is that whenever \( m \) is significantly larger than \( n \), a permutation matrix exists such that the lower bound is met exactly. The second is that given an optimal permutation matrix of \( k - 1 \) columns, one can permanently fix these columns and find a \( k \)th column that satisfies

\[ |m(m^2 - 1)r_{ik}/6| = |m(m^2 - 1)/6 - \sum D^2_0| = \begin{cases} 1 & \text{if } m = 2 + 4l, \ l \in \mathbb{Z}_1^+, \\ 0 & \text{otherwise}, \end{cases} \]

for all \( i = 1, \ldots, k - 1 \). By the assumption that the matrix of \( (k - 1) \) columns is optimal, it follows that adding such a \( k \)th column to the other \( (k - 1) \) columns provides an optimal permutation matrix of \( k \) columns.

As stated in our earlier work, empirical testing not only strongly supported the conjecture that one can sequentially solve the \( n \) dimensional problem by solving \( n - 1 \) significantly smaller problems, but made apparent two additional features of the problem that are interesting from the optimization viewpoint. They are that there exist for a given \( m \) and \( n \) many optimal solutions from which to choose, and that the optimal solutions are checkably optimal in linear time (i.e. given an optimal permutation matrix, proving it is optimal is trivial).

To develop alternative formulations, we use the objective function lower bound of \( (\binom{n}{k})6/[m(m^2 - 1)] \) when \( m = 2 + 4l \) for some nonnegative integer \( l \), and zero otherwise, and the above conjecture.

Suppose we have a solution to the \((k - 1)\)-dimensional problem and wish to use this solution to obtain a solution to the \(k\)-dimensional problem. Let \((p^1, p^2, \ldots, p^k)\) denote the corresponding column permutation vectors, and define

\[ x_{ij} = \begin{cases} 1 & \text{if the } i \text{th element of column } k \\
& \text{is assigned value } j, \\ 0 & \text{otherwise.} \end{cases} \]

To ensure that column \( k \) is a permutation of numbers \( 1 \ldots m \), we add the assignment constraints:

\[ \sum_i x_{ij} = 1, \quad j = 1, \ldots, m, \]
\[ \sum_j x_{ij} = 1, \quad i = 1, \ldots, m. \]

We see that the positive elements of an \( x \)-solution define a \( k \)th column. We will henceforth interchangeably use the terms "an \( x \)-solution" and "the \( k \)th column defined by the positive elements of \( x \)".
There are \((k-1)\) additional constraints of the following form:

\[
d_{t,k}^+ - d_{t,k}^- = m(m^2 - 1) / 6 - \sum_{i=1}^{m} \sum_{j=1}^{m} (p_i^t - j)^2 x_{ij}, \quad t = 1, \ldots, k - 1,
\]

where \(p_i^t\) is the \(i\)th entry of the column permutation vector \(p^t\). With these constraints, we implicitly fix the \((k-1)\) previously found column permutations.

The formulation defined thus far with objective function

\[
\min \sum_{i=1}^{k-1} (d_{ik}^+ + d_{ik}^-)
\]

is a general formulation for finding a \(k\)th column permutation, having fixed the \((k-1)\) column permutations that minimize \(\sum_{i=1}^{k-2} \sum_{j>i} |r_{ij}|\). If we assume, as the empirical evidence strongly supports, that the conjecture is true, then there exists a \(k\)th column that meets the lower bound for \(|r_{ij}|, i = 1, \ldots, k - 1\). Hence, there exists a solution to the assignment constraints that generates a \(k\)th column satisfying

\[
|m(m^2 - 1)r_{ik}/6| = |m(m^2 - 1)/6 - \sum D_{0}^2 | = \begin{cases} 1 & \text{if } m = 2 + 4l, \ l \in Z^*_1, \\ 0 & \text{otherwise}, \end{cases}
\]

for all \(i = 1, \ldots, k - 1\). To incorporate this into the formulation, we require that \(d_{ik}^+\) and \(d_{ik}^-\), \(t = 1, \ldots, k - 1\), be binary variables. For \(m \neq 6 + 4l\), any solution that obtains the lower bound must have \(d_{ik}^+ + d_{ik}^- = 0\). If, however, \(m = 2 + 4l\), we can conclude that \(d_{ik}^+ + d_{ik}^- = 1, \ t = 1, \ldots, k - 1\). In either case, the problem can be restated as a feasibility problem with no objective function. We shall refer to this feasibility assignment problem with side equations as FASE.

We note that since the feasibility solutions to the constraint set of FASE are optimal permutation columns for \((m \times k)\) MCLHS problems, the objective function value \(Z_{kp}\) for all feasible FASE solutions is zero. That is, the value of \(Z_{kp}\) cannot direct the search for feasibility. As we shall see in the next section, this will have ramifications for relaxations of FASE.

Assuming the conjecture holds, rather than solving one large APSEC program with \(m^n + n(n-1)\) variables and \(nm + \binom{n}{2}\) constraints, one could solve a sequence of smaller two-dimensional FASE problems with at most \(m^2 + 2k\) variables and \(2m + k\) constraints \((2 \leq k < n)\). In the next section, we consider a Lagrangian relaxation of these subproblems that will be used in the exact solution approaches.

3. Exchange heuristics and Lagrangian relaxation approaches to FASE

We shall now consider a Lagrangian relaxation of both FASE and an extension of FASE that we will introduce shortly. To simplify the discussion, the \(m \times m^2\) matrices corresponding to the constraints \(\sum_{i=1}^{m} x_{ij} = 1, \ j = 1, \ldots, m\) and \(\sum_{j=1}^{m} x_{ij} = 1, \ i = 1, \ldots, m\)
i = 1, ..., m, will henceforth be denoted by $A^1$ and $A^2$, respectively. Periodically, the matrix formed by the $2m \times m^2$ assignment constraints of both $A^1 x = 1$ and $A^2 x = 1$ taken together will simply be written as $A$. For $m \neq 6 + 4l$, $A^3$ will be a $(k - 1) \times m^2$ matrix corresponding to the knapsack equations $\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} x_{ij} = m(m^2 - 1)/6$, $t = 1, \ldots, k - 1$. When $m = 2 + 4l$, $A^3$ will be a $(k - 1) \times [m^2 + 2(k - 1)]$ matrix corresponding to the knapsack equations

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} x_{ij} - d_{ik}^+ + d_{ik}^- = m(m^2 - 1)/6, \quad t = 1, \ldots, k - 1,$$

where $w$ will denote the $[m^2 + 2(k - 1)]$-dimensional solution vector of $x$ and $d$.

The unrestricted Lagrangian multipliers needed from relaxing constraints associated with $A^1$, $A^2$, or $A^3$ will be suitably dimensioned $\mu^1$, $\mu^2$, and $\mu^3$. In addition, $B^n$ will denote the set of $n$-dimensional binary vectors.

To begin, we note that since FASE is a feasibility problem, its optimal value for the linear programming (LP) relaxation bound $Z_{LP}$ will be zero for all values of $m$. Although there is no gap between $Z_{LP}$ and $Z_{FP}$, and the solutions satisfy the assignment and knapsack constraints, they are, in general, fractional.

When $m = 2 + 4l$, we could strengthen the FASE formulation by appending the constraints

$$d_{ik}^+ + d_{ik}^- = 1, \quad t = 1, \ldots, k - 1.$$

This problem, hereafter referred to as FASEM, will not guarantee integrality for the $d$-variables, but will at least eliminate a sizeable number of fractional solutions otherwise allowed. We will henceforth presume that when $m = 2 + 4l$, we address FASEM and define $A^4$ the $(k - 1)$ matrix associated with the new constraints $d_{ik}^+ + d_{ik}^- = 1, t = 1, \ldots, k - 1$. The unrestricted Lagrangian multiplier needed from relaxing constraints associated with $A^4$ will be a suitably dimensioned $\mu^4$.

Since $Z_{LP} = Z_{FP}$ for all $m$, we consider potential Lagrangian relaxations of both FASE and FASEM not from the traditional perspective of choosing one that will achieve the strongest bound on the objective function value, but rather select one that we believe will have the greatest likelihood of obtaining integer solutions without too much computational difficulty.

With this in mind, the most apparent relaxation is found by relaxing the constraints $A^3 x = m(m^2 - 1)/6$ ($A^3 w = m(m^2 - 1)/6$) for FASE (FASEM). Thus, for the FASE problems, the total unimodularity of $A$ ensures that the LP relaxations of the resulting subproblems will generate integral solutions whose corresponding columns will, by definition, be permutations of the numbers 1 to $m$. These permutations may not be feasible to the relaxed constraints $A^3 x = m(m^2 - 1)/6$.

When the feasibility problem under consideration is FASEM, we see that relaxing the constraints $A^3 w = m(m^2 - 1)/6$ results in a Lagrangian subproblem:
Minimize \( \mu^3 \frac{m(m^2 - 1)}{6} - A^3 w \)

subject to \( (x \in B^{m^2}, d \in B^{2(k-1)} | Ax = 1, A^4 d = 1) \).

Decomposing the objective function, we have

Minimize \( \mu^3 \frac{m(m^2 - 1)}{6} - \mu^3 A^{3x} x - \mu^3 A^{3d} d, \)

where \( A^{3x} \) and \( A^{3d} \) are the first \( m^2 \) and remaining two columns of \( A^3 \) associated with \( x \) and \( d \), respectively. Since the constraints are disjoint, given a \( \mu^3 \), we can determine an optimal \( \vec{d} \) as follows:

\[
\begin{align*}
\text{if} \quad \mu^3 > 0, & \quad \vec{d}_{ik}^- = 1 \text{ and } \vec{d}_{ik}^+ = 0 \\
\text{else} & \quad \vec{d}_{ik}^+ = 1 \text{ and } \vec{d}_{ik}^- = 0.
\end{align*}
\]

Substituting \( \vec{d} \) into the objective, we now address an assignment problem with the objective function

Minimize \( \mu^3 \left[ (m(m^2 - 1)/6 - A^{3d} \vec{d}) - A^{3x} x \right] \).

The LP relaxation of this subproblem can again be solved to yield integral solutions that correspond to permutations of the numbers 1 to \( m \). Thus, while the discussion below will assume that \( m \neq 2 + 4l \) (i.e. only the feasibility problem FASE is considered), we note that it can be extended for all cases of \( m \).

The Lagrangian relaxation approach described below will be used only when a basic exchange heuristic fails to find an optimal solution at some iteration. This heuristic is also used within the Lagrangian techniques, as we will explain later. For this reason, an outline of the heuristic follows.

Define \( p \) to be a vector of length \( m \) that is a permutation of the numbers 1 to \( m \), where \( p_i \) is the \( i \)th element of the vector. Since we are seeking \( n \) such vectors, we will henceforth denote \( p^k \) to be the \( k \)th vector and \( p^k_i \) the \( i \)th element of vector \( p^k \). The objective value that would result from an interchange of \( p^k_i \) and \( p^k_j \) will be denoted as \( Z_{new}^{ij} \). The overall approach of the heuristic is:

**Step 0:** *Initialization*

0.1 Set \( k = 1 \).
0.2 Set \( p^1_i = i \) for \( i = 1 \ldots m \).

**Step k:**

k.1 Set \( k = k + 1 \)

k.2 Determine an initial permutation for column \( k \).

k.3 Exchange elements of column \( k \) to find a permutation of numbers 1 to \( m \) such that \( \sum_{i=1}^{k-1} |m(m^2 - 1)/6 - \sum_{i=1}^{m} (p^k_i - p^1_i)^2| \) is as close to zero or \( (k - 1) \) as possible, depending on \( m \).
In HHY [7], we have shown that this heuristic obtains optimal solutions for cases when \( n = 2 \) and \( n = 3 \), with \( m > 8 \). When \( n > 3 \), the heuristic found optimal or near-optimal solutions for all cases of \( m \) tested. For further details of the heuristic and results from computational testing, see HHY [7].

We use this heuristic until we arrive at a step \( k \) where the heuristic fails to find an optimal solution. We fix the \( k - 1 \) columns for which the heuristic found optimal column permutations and begin the exact solution approach with a search for the \( k \)th column. The Lagrangian procedure continues until \( n \) column permutations are identified. Thus, the exact approaches discussed are also based on the assumption that the conjecture holds, and such \( k \)th columns can be iteratively found. A general overview of the method for solving the entire MCLHS problem is:

**Step 1.0.** Set \( k = 2 \).

**Step 1.1.** Run the MCLHS heuristic to find \( k \) optimal column permutations.
If the MCLHS heuristic is successful, go to step 1.2.
Else go to step 1.3.0.

**Step 1.2.** Set \( k = k + 1 \).
If \( k = n \), then RETURN.
Else go to step 1.1.

**Step 1.3.0.** Find an optimal \( k \)th column permutation using an exact solution procedure.

**Step 1.3.1.** Fix the \( k \)th column, and set \( k = k + 1 \).

**Step 1.3.2.** Go to step 1.1.

All Lagrangian-based solution approaches require the determination of initial multipliers for the Lagrangian dual and then a procedure that is guaranteed to have the Lagrangian converge to the optimal Lagrangian solution value. FASE's lack of an objective function creates difficulties in that there is no basis for setting initial multiplier values and the only things driving the search (direction) are the Lagrange constraints which make up the objective function. Hence, the usual multiplier adjustment and updating schemes that exploit relationships between the original costs and penalties incurred from infeasibilities of the relaxed constraints cannot be used.

A further complicating factor in determining successive values for multipliers is that since the FASE constraints are equations, all multipliers are unrestricted in value. It is therefore likely that the objective value, dependent only upon the infeasibilities of the relaxed constraints, will alternate between positive and negative values. In the two approaches that follow, the search for optimal multipliers will be performed using a subgradient method.

While both solution approaches are based on the relaxation discussed above, we consider first a traditional Lagrangian method. Suppose we are given \( k - 1 \) column permutations, and we wish to determine a \( k \)th permutation that optimizes the \( m \times k \) MCLHS problem. Define \( \mu^3_{ur} \) as the unrestricted multiplier needed from relaxing the
rth constraint of \( A^3x = m(m^2 - 1)/6 \) at iteration \( v \) below, and \( \alpha_v \) to be the subgradient stepsize. The general approach is:

**Step 0: Initialization**

0.0. Execute the MCLHS heuristic to find a \( k \)th column permutation. If this \( k \)th column permutation is optimal, then RETURN. Else go to step 0.1.

0.1. Set \( v = 1 \).

0.2. Initialize \( \mu_{1t}^3 \) for \( t = 1, \ldots, k - 1 \).

0.3. Initialize \( \alpha_1 = 2.0 \).

**Step \( v \):**

\( v.1 \). Solve subproblem for \( x_{new} \)

\[
\text{minimize} \quad z = \mu_v^3 [m(m^2 - 1)/6 - A^3x]
\]

subject to \( A^1x = 1, \ A^2x = 1, \)

\( x \in B^{m^2} \).

\( v.2 \). If \( A^3x_{new} = m(m^2 - 1)/6 \), then RETURN. Else go to step \( v.3 \).

\( v.3 \). Update multipliers and stepsize.

\( v.3.1 \). General approach for \( \mu_{v+1}^3 \):

\[
\mu_{v+1}^3 = \mu_v^3 + \alpha_v [M - A^3x_{new}]/\|M - A^3x_{new}\|^2,
\]

where \( M = m(m^2 - 1)/6 \).

\( v.3.2 \). General approach for \( \alpha_v \):

If the objective value \( z \) has not improved after some preset number of iterations \( L \), then

\[
\alpha_{v+1} = \alpha_v/2.
\]

Otherwise, \( \alpha_{v+1} = \alpha_v \).

\( v.3.3 \). \( v = v + 1 \),

go to step \( v.1 \).

In an attempt to compensate for the lack of an objective function, the initial setting of \( \mu_0^3 \) is based upon the near-optimal solution generated by the MCLHS heuristic. Letting \( \alpha_0^3x = m(m^2 - 1)/6 \) be the \( r \)th row of \( A^3x = m(m^2 - 1)/6 \), we define

\[
IFS_r = m(m^2 - 1)/6 - \alpha_0^3x.
\]
Further, we identify the row index, $t_{\text{max}}$, that satisfies

$$t_{\text{max}} = \arg \max_t (|\text{IFS}_t|).$$

Now, for all rows such that $\text{IFS}_t = 0$, $\mu^3_{it}$ is set to zero. For all other rows of $A^3x = m(m^2 - 1)/6$ that are not met exactly, $\mu^3_{it}$ is set to

If $t = t_{\text{max}}$, then

$$\mu^3_{it} = K$$

Else

$$\mu^3_{it} = (\text{IFS}_t / |\text{IFS}_{t_{\text{max}}}|) K,$$

where $K$ is a preset parameter between 1 and 10.

We use the stepsizes consistent with that suggested in the literature (see, for instance, Held et al. [8]) and use an update of $\alpha_{u+1} = \alpha_u / 2$.

As theory predicts, this method converged to the optimal solution value. However, in no case did the method provide a feasible integer solution to the problem. In essence, in each case the multipliers converged to zero, thereby supplying a value of zero for the objective function value (the correct value for the integer solution), but without forcing the violated constraint to zero.

While the approach discussed above appeared to have promising theoretical properties, computationally it was disappointing because it failed to identify exact solutions to FASE.

As a result, we were prompted to consider employing our MCLHS heuristic, which obtains near-optimal and often optimal solutions, in an interactive way with a Lagrangian approach. Thus, once a Lagrangian solution is found, the MCLHS heuristic is executed in the hope of obtaining an improved permutation that satisfies the correlation knapsack equations $A^3x = m(m^2 - 1)/6$.

Employing the Lagrangian approach that is based on the relaxation where $A^3x = m(m^2 - 1)/6$ are relaxed, the initialization for this hybrid approach is identical to that of the approach discussed above. The principal modification is at step $u$, where we now have:

**Step $u$:**

**u.1.** Solve subproblem for $x_{\text{new}}$

minimize $z = \mu^3 [m(m^2 - 1)/6 - A^3x]$

subject to $A^1x = 1$, $A^2x = 1$,

$$x \in B^{m^2}.$$

**u.2.** If $A^3x_{\text{new}} = m(m^2 - 1)/6$, then RETURN.

Else go to step **u.3**.
3. If \( x_{\text{new}} \) is the same solution as that found by
iteration \((v - 1)\), then go to step 4.
Else
Execute the MCLHS heuristic to improve solution \( x_{\text{new}} \).
If knapsack equations are satisfied, then RETURN.
Else go to step 4.

4. Update multipliers and stepsizes.

4.1. General approach for \( \mu_{v+1}^3 \):
\[
\mu_{v+1}^3 = \mu_v^3 + \alpha_v [M - A^3 x_{\text{new}}] / \| M - A^3 x_{\text{new}} \|^2,
\]
where \( M = m(m^2 - 1)/6 \).

4.2. General approach for \( \alpha_v \):
If the objective value \( z \) has not improved after some preset number
of iterations \( L \), then
\[
\alpha_{v+1} = \alpha_v / 2.
\]
Otherwise, \( \alpha_{v+1} = \alpha_v \).

4.3. \( v = v + 1 \),
then go to step 1.

Thus, we have used the Lagrangian relaxation in an attempt to provide an
optimal integer solution and not to find a bound on the optimal solution value, the
most common use for this technique. Rather, we attempted to find an algorithm that
would identify a permutation having a specified value (i.e. the solution value is
known). We can, therefore, look at the Lagrangian portion of this hybrid approach
as a mechanism for providing starting points for our heuristic search technique.
Every time the multipliers changed, we re-solved an assignment problem. If the
assignment solution was one not previously considered, the heuristic was executed
to improve this solution to be an optimal one. Thus, the Lagrangian procedure
provided the exchange heuristic solutions close to optimality and the heuristic could
then search for an optimal solution.

In HHY [7], we presented computational results for the MCLHS heuristic. We
stated that for all two-dimensional cases of \( m = 9, \ldots 1000 \) and three-dimensional
cases of \( m = 9, \ldots 300 \), optimal solutions were identified in relatively short time.
When \( n = 4 \) was tested for values \( m = 12, \ldots 101 \), the heuristic fixed the first three
optimal column permutations and attempted to find a fourth column that minimized
\( |r_{12}| + |r_{24}| + |r_{34}| \). In the worst case, the heuristic identified near-optimal column
permutations that met exactly the lower bound for two of these three rank correlations.

Implementing the approach discussed above, we found that the algorithm
successfully found optimal solutions for all four- and five-dimensional cases when
\( m = 10, \ldots 180 \) and \( m = 50, \ldots 100 \), respectively. All problems were tested on an IBM
C.M. Harris et al., Using integer programming techniques

Table 1
Hybrid solution approach, computational results for $n = 4$.

<table>
<thead>
<tr>
<th>Size of $m$</th>
<th>No. HEUR probs</th>
<th>LAG probs</th>
<th>Max LAG steps</th>
<th>Tot LAG steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-20</td>
<td>11</td>
<td>10</td>
<td>15</td>
<td>46</td>
</tr>
<tr>
<td>21-40</td>
<td>20</td>
<td>15</td>
<td>19</td>
<td>106</td>
</tr>
<tr>
<td>41-60</td>
<td>20</td>
<td>10</td>
<td>15</td>
<td>76</td>
</tr>
<tr>
<td>61-80</td>
<td>20</td>
<td>10</td>
<td>13</td>
<td>74</td>
</tr>
<tr>
<td>81-100</td>
<td>20</td>
<td>5</td>
<td>10</td>
<td>24</td>
</tr>
<tr>
<td>101-120</td>
<td>20</td>
<td>11</td>
<td>12</td>
<td>42</td>
</tr>
<tr>
<td>121-140</td>
<td>20</td>
<td>15</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>141-160</td>
<td>20</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>161-175</td>
<td>20</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>176-180</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2
Hybrid solution approach, computational times for $n = 4$.

<table>
<thead>
<tr>
<th>Size of $m$</th>
<th>Total time</th>
<th>Avg time</th>
<th>Tot HEUR time</th>
<th>Avg HEUR time</th>
<th>Tot LAG time</th>
<th>Avg LAG time</th>
<th>HEUR time for LAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-20</td>
<td>9.91</td>
<td>0.90</td>
<td>0.21</td>
<td>0.21</td>
<td>9.7</td>
<td>0.97</td>
<td>7.47</td>
</tr>
<tr>
<td>21-40</td>
<td>49.0</td>
<td>2.45</td>
<td>2.31</td>
<td>0.46</td>
<td>46.72</td>
<td>3.11</td>
<td>32.55</td>
</tr>
<tr>
<td>41-60</td>
<td>74.23</td>
<td>3.71</td>
<td>14.81</td>
<td>1.48</td>
<td>59.42</td>
<td>5.94</td>
<td>34.49</td>
</tr>
<tr>
<td>61-80</td>
<td>152.68</td>
<td>7.64</td>
<td>19.97</td>
<td>1.99</td>
<td>132.71</td>
<td>13.27</td>
<td>74.32</td>
</tr>
<tr>
<td>81-100</td>
<td>174.99</td>
<td>8.75</td>
<td>59.44</td>
<td>4.57</td>
<td>115.55</td>
<td>16.51</td>
<td>39.61</td>
</tr>
<tr>
<td>101-120</td>
<td>273.70</td>
<td>13.68</td>
<td>140.64</td>
<td>9.38</td>
<td>133.06</td>
<td>26.61</td>
<td>56.03</td>
</tr>
<tr>
<td>121-140</td>
<td>512.78</td>
<td>25.64</td>
<td>111.22</td>
<td>10.11</td>
<td>401.56</td>
<td>44.62</td>
<td>156.94</td>
</tr>
<tr>
<td>141-160</td>
<td>493.61</td>
<td>24.68</td>
<td>116.27</td>
<td>12.74</td>
<td>302.58</td>
<td>60.52</td>
<td>116.27</td>
</tr>
<tr>
<td>161-174</td>
<td>940.73</td>
<td>67.2</td>
<td>235.44</td>
<td>23.54</td>
<td>705.29</td>
<td>176.32</td>
<td>455.38</td>
</tr>
<tr>
<td>176-180</td>
<td>132.57</td>
<td>26.52</td>
<td>132.57</td>
<td>26.52</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

RS6000, model 540 workstation, and assignment subproblems solved by SEMI, a FORTRAN code designed to solve sparse assignment problems. To conclude this section, we present the computational results of this implementation.

In tables 1 and 3, the problems tested are separated into those that were solved by the heuristic and those that required the approach using both Lagrangian relaxation and the MCLHS heuristic. For the latter problems, the fifth column indicates, for each group of problems tested, the maximum number of Lagrangian steps needed to optimally solve any problem in the group. The entries of the last column are the total number of Lagrangian steps required to solve all problems in a group.
In tables 2 and 4, computational times associated with the implementation are given for \( n = 4 \) and \( n = 5 \), respectively. Total time is again divided between the total time needed for the solution of problems requiring only the MCLHS heuristic and the total time needed for solving problems that required both the heuristic and Lagrangian procedures. The final column of these tables gives the time actually spent in the MCLHS heuristic for the latter set of problems that required the complete approach to solve to optimality. From this, we see that the majority of effort for the approach described above is spent in the MCLHS heuristic searching to bring an assignment solution to optimality.

4. Summary and conclusions

Latin hypercube sampling is of interest not only to statisticians, but to a variety of researchers in fields such as operations research, economics, and engineering, because such researchers are faced with experimentation that is often time-consuming and/or expensive to perform on large samples or where it may not be possible to perform sampling on much of the population under study. Often, many econometric and simulation modeling efforts require the repeated execution of a computer model to determine the sensitivity of the outputs to perturbations of the input parameters. These models are often expensive to run, thereby limiting the total sampling (runs) possible. Harris [5] offers an overview and introduction of the use of LHS design for evaluation of the sensitivity of a model’s outputs to changes in the input values. Iman
and Conover [9] use LHS designs in the sensitivity analysis of the Sandia Waste Isolation Flow and Transport (SWIFT) computer program. This program's objective is to determine the potential escape of radio-nuclides from a nuclear waste site and their migration from the subsurface to the surface environment. In order to simulate the migration of radio-nuclides in large systems over time, the program requires extensive computing and, thus, the number of runs possible is limited. In seeking the maximum amount of information about SWIFT's outputs from a limited number of simulation runs, Iman and Conover use standard LHS plans to estimate the mean and variance of the model output. In this study, input variables such as the time to ground-water contact and properties of rock types near the waste sites were believed to be independent. The authors did not consider whether the standard LHS plans accurately represented this independence.

In 1984, Iman and Conover [10] considered the issue of the dependence of SWIFT's input variables and found non-zero rank correlations among them. Their previous standard plans met neither an independence assumption nor accurately reflected the true rank correlation of the variables. In their paper, they attempted to develop a procedure that would provide an LHS design matrix meeting their rank correlation requirements. Concluding that "it does not appear possible to find a transformation matrix which results in the target rank correlations matrix", the authors proposed a heuristic approach to finding the target matrix. We applied the heuristic described in this paper to two illustrations given in this paper. The approach works "almost instantaneously" and provided correlations that very closely matched the required correlations of zero. (For a 15-variable, 100-level case, the average absolute error was 0.000378, with the largest deviation being 0.005866.)

Other examples that could benefit from the technology of this paper are the samplings performed in Chapman and Yakowitz [2] which deal with the break-even analysis of a waste management model, and Harris [4] which examined the sensitivity of outputs to small perturbations in inputs for a large energy-resource model used by the Department of Energy.

We note that the solution obtained for an \((n, m)\)-MCLHS problem is valid regardless of the underlying application. Thus, although the modeling performed by each of the above cited authors is quite different in subject matter and scope, each would use the same solution to the MCLHS design problem, if each wanted to determine the effects of altering, for example, 10 input parameters and was limited to performing 500 runs of their respective models. Thus, for each \((m, n)\)-pair, one optimization problem is solved and the results recorded so that anyone needing to perform an MCLHS of size \((m, n)\) could look up the optimal design.

Although it is true that the most direct IP formulation of the MCLHS problem grows exponentially (with \(O(m^n)\) variables) in size and belongs to the class of NP-complete problems, that is no reason for not attempting to obtain good, and even optimal, solutions to specific instances of these problems. Combinatorial optimization techniques have regularly solved important, very large problems in this NP-class and
we have provided another problem class where examining the structure of the problem gives insights to solution approaches.

By exploiting both the objective function lower bound and the fact that is was possible to obtain solutions to higher-dimensional problems from solutions to lower-dimensional instances, we created an algorithm that could be solved iteratively by using the strength of reformulation, heuristics and relaxation techniques. This hybrid approach works well specifically because there are many feasible integer solutions and the heuristic is fast in finding an optimal solution when a number of different starting points are provided.

Acknowledgements

Special thanks to Jeffery Kennington and Zhiming Wang for providing the SEMI software used in this work. We also wish to thank the referees for their careful reading and editorial suggestions. The reorganization of the paper suggested by one of the referees significantly improved the flow of the presentation.

References