응용통계연구 제 7권 1호
1994년 3월 pp. 83 - 102

Design and Analysis of Computer Experiments with An Application to Quality Improvement[1)]

Jung-Wook Sim, Jeong-Soo Park, Jong-Sung Bae[2)]

ABSTRACT

Some optimal designs and data analysis methods based on a Gaussian spatial linear model for computer simulation experiments are considered. For designs of computer experiments, Latin-hypercube designs and some optimal designs are combined. A two-stage computational (2-points exchange and Newton-type) algorithm for finding the optimal Latin-hypercube design is presented. The spatial prediction model, which was discussed by Sacks, Welch, Mitchell and Wynn (1989) for computer experiments, is used for analysis of the simulated data. Moreover, a method of contructing sequential (optimal) Latin-hypercube designs is considered. An application of this approach to the quality improvement and optimization of the integrated circuit design via the main-effects plot and the sequential experimental strategy is presented.

1. Introduction

Modern scientific researchers often use complex computer simulation codes for theoretical investigations because physical experiments are too expensive, as in the case of nuclear reactor experiments, or simply impossible, as in weather modeling. Heat transfer in engineering structure, meteorological phenomena, plasma behavior in physics and global economic activity, for examples, are modeled mathematically often in the form of large or complicated systems of differential equations, and computer programs are written to evaluate the outputs of interest that result from specified inputs.

In a "computer simulation experiment", observations are made on an output $Y(x)$ by running a computer code at various choice of input variables, x. One feature of a computer simulation experiment, different from a physical experiment, is that the output is often deterministic-the response is observed without measurement error. This calls for new distinct techniques useful in modeling deterministic systems.

Since simulation codes are often computationally very expensive to run, a careful selection of inputs is necessary for efficient analysis of the data, which leads to a design problem for the computer simulation experiment.

---

1) This paper was supported by NON-DIRECTED RESEARCH FUND, Korea Research Foundation, 1992. Original version of this paper was presented at the "Workshop on The Experimental Designs for Quality Improvements" held at Seoul National University, June 1992.

2) Department of Statistics, Chonnam National University 300 Yongbong-dong, Buk-gu Kwangju 500-757, Korea

This paper deals with design and analysis of computer simulation experiments. For designs of computer experiments, Latin-hypercube designs and some optimal designs based on a spatial prediction model are combined in section 2. A two-stage algorithm for finding the optimal Latin-hypercube design is described in section 3. The spatial prediction model, which was discussed by Sacks, Welch, Mitchell and Ylvisaker (1989) for computer experiments, is also used for analysis of the simulated data. In section 5, an application of this approach, which is a mixture of response surface methodology and spatial statistics (Cressie, 1991), to the quality improvement of the integrated circuit design is presented. A method of constructing sequential (optimal) Latin-hypercube designs is considered in section 6.

2. Designs for Computer Experiments and A Spatial Model

McKay, Beckman and Conover (1979), and Iman and Conover (1980) proposed Latin-hypercube designs (or Latin-hypercube sampling, abbreviatedly Lhd). They and Stein (1988) showed that Lhd is generally useful and efficient for computer experiments. Some advantages are that it is computationally cheap to construct, flexible for various input distributions and covers the design region well without replications.

An optimal design for computer experiments which minimizes the integrated mean squared error of prediction (IMSE) was introduced by Sacks, Schiller and Welch (1989), and by Sacks, Welch, Mitchell and Wynn (1989, abbreviatedly SWMW). Moreover, Shewry and Wynn (1987), and Currin, Mitchell, Morris and Ylvisaker (1989, CMMY) investigated an entropy maximizing design for computer experiments.

These optimal designs were illustrated to be more efficient, in the sense of prediction at the untried input sites, than factorial designs and Lhd's. However, the computational cost of finding optimal designs is high, especially when the number of input variables d and/or design points n are large. Thus a design which is cheap to construct, geometrically appealing and optimal in some sense has been sought (Section 5 in CMMY, and Easterling, 1989, in the discussion on SWMW).

In this section, by combining the advantages of both Lhd and optimal design, we consider optimal Latin-hypercube design (OLhd) which is a Lhd optimizing a given criterion such as IMSE or entropy.

Model-based Optimal Designs: Sacks, Schiller and Welch (1989, abbreviatedly SSW) used a spatial stochastic model which treats the computer response $Y(\underline{x})$ as a realization of a random function superimposed on a regression model,

$$Y(\underline{x}) = \sum_{j=1}^{k} \beta_j f_j(\underline{x}) + Z(\underline{x}), \tag{2.1}$$

where f's are known functions and β's are unknown regression coefficients.

The random process $Z(.)$ is assumed to be a Gaussian process with mean zero and

covariance

$$Cov(\underline{t}, \underline{u}) = \sigma_z^2 \exp\{-\sum_{j=1}^{d} \theta_j |t_j - u_j|^{\alpha}\}, \quad 0 < \alpha \leq 2, \tag{2.2}$$

between $Z(\underline{t})$ and $Z(\underline{u})$, for two d-dimensional inputs $\underline{t} = (t_1, ..., t_d)$ and $\underline{u} = (u_1, ..., u_d)$, where σ_z^2 is the process variance. The non-negative parameter θ determines the correlation structure of Z: small θ reflects large correlations between nearby observations while large θ reflects small nearby correlations (see Figure 1 for the effect of θ to the prediction surface). For the smoothness of the realization of Z and the convenience of the computation of IMSE (analytically computable), we usually set $\alpha = 2$.

Once a covariance function and its parameters are specified, one can predict $Y(x)$ based on the model (2.1) using the observations $y = Y(s)$. For the prediction formula, define the $n \times n$ correlation matrix V and $n \times k$ matrix F by

$$V = \frac{1}{\sigma_z^2} [Cov(s_i, s_j)]_{\substack{1 \leq i \leq n, \\ 1 \leq j \leq n}} \tag{2.3}$$

and

$$F = [f_l(s_i)]_{\substack{1 \leq i \leq n, \\ 1 \leq l \leq k}}$$

where $(s_i, ..., s_n)$ are the design sites (note that each design site is a d-dimensional vector). Here V is the correlation matrix determined by observation sites (or design sites), and F is so-called the design matrix.

For any prediction site x, the $n \times 1$ vector v_x and $k \times 1$ vector f_x are defined by

$$\begin{aligned} v_x' &= [Cov(s_1, x), ..., Cov(s_n, x)]/\sigma_z^2, \\ f_x' &= [f_1(x), ..., f_k(x)], \end{aligned} \tag{2.4}$$

respectively. Here v_x is a correlation vector between design sites s_i's and a prediction site x. Then the best linear unbiased predictor (BLUP) of $Y(x)$ given the observation vector y is (see SWMW or Ripley, 1981, pp. 44-58)

$$\begin{aligned} \hat{Y}(x) &= [v_x' \; f_x'] \begin{bmatrix} V & F \\ F' & 0 \end{bmatrix}^{-1} \begin{bmatrix} y \\ 0 \end{bmatrix} \\ &= f_x' \hat{\beta} + v_x' V^{-1}(y - F\hat{\beta}), \end{aligned} \tag{2.5}$$

where $\hat{\beta} = (F' V^{-1} F)^{-1} F' V^{-1} y$ is the generalized least squares estimator of β. Thus

the prediction of $Y(x)$ in this model is interpreted as a sum of the generalized least squares prediction of $Y(x)$ and the quantity which is related to the observations y and the correlation v_x. This prediction (2.5) is called as "universal Kriging" in spatial statistics literature (See Cressie, 1991 or Ripley, 1981). Here the matrix $\begin{bmatrix} V & F \\ F' & 0 \end{bmatrix}$ is of course invertible because V is a positive definite correlation matrix.

The mean squared error of prediction is

$$MSE(x) = MSE(\hat{Y}(x)) = E_\theta[\hat{Y}(x) - Y(x)]^2, \qquad (2.6)$$

and the normalized mean squared error, $mse(x) = MSE(x)/\sigma_z^2$ is

$$mse(x) = 1 - [v_x' \; f_x'] \begin{bmatrix} V & F \\ F & 0 \end{bmatrix}^{-1} \begin{bmatrix} v_x \\ f_x \end{bmatrix}. \qquad (2.7)$$

The main advantage of the model (2.1) is that the prediction surface interpolates the observations because the predictor $\hat{Y}(s_i)$ at a design point s_i has $mse(s_i) = 0$, i.e., $\hat{Y}(s_i) = Y(s_i)$ (see Ripley, 1981, pp. 44-58, or Park and Park, 1991). This is one of the reasons why the model is used for deterministic data analysis of computer experiments. Figure 1 illustrates the interpolating property of the prediction (2.5) and the effects of θ to the prediction surface.

For a given "weight measure" mu on a support set Q, the (normalized) integrated mean squared error (IMSE) is

$$IMSE = \int_Q mse(x) \, d\mu(x) \; . \qquad (2.8)$$

Note that neither mse nor IMSE depend on the data y nor on the unknown parameters β and σ_z^2. Then the IMSE is only a function of the design sites. This makes it possible to design an experiment before taking the data, i.e., to select the observation sites (or the design sites) which optimize some criteria such as the IMSE or the maximum mse (see SWMW or SSW on this direction).

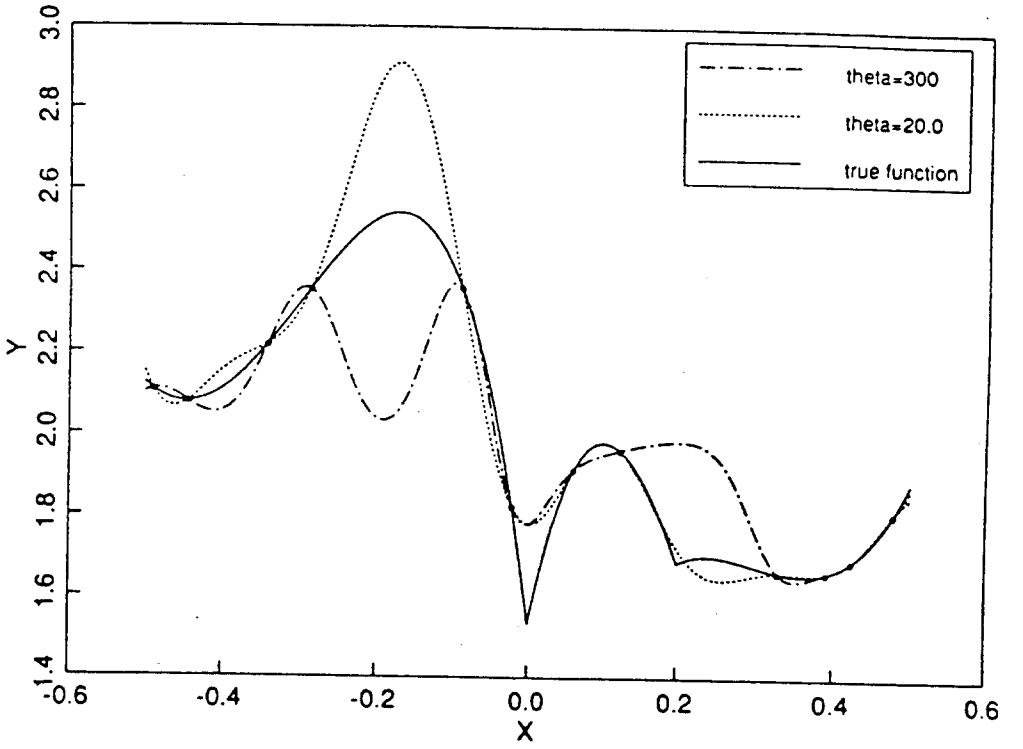

Figure 1: The effect of θ in prediction and the interpolating property at the observation sites are illustrated with the prediction surfaces for different θ's, and a fixed α=2. Twelve responses (dots) from the true function (the solid line) are observed at the randomly selected design sites. The covariance parameter θ works as a smoothing parameter: smaller θ gives more smooth prediction while large θ gives more fluctuations with tendency of backing to the response mean.

In using the model (2.1) for the design and analysis of computer experiments, input variables are usually shifted and rescaled so that the design and the prediction region are in $[-.5, .5]^d$ because we need a common scale which removes the effect of different scales of the input variables. Thus it is not necessary to use only [-.5,.5] scale. One can of course use [0,1] or [-1,1] scale for example.

Shewry and Wynn (1987), and CMMY investigated the maximal entropy design in which the unsampled population has minimum variability conditional on the selected design. In the model (2.1) for fixed β, maximum entropy criterion is equivalent to maximizing the determinant of the correlation matrix V in (2.2), which is called a D-optimality.

Even though these optimal designs are efficient and useful for computer experiments with prediction model (2.1), the cost of design construction is computationally expensive. This is the case especially when the sample size (n) and/or the number of input variables (d) are large, because it is a $n \times d$-dimensional optimization problem.

Since the response function in computer experiments is often deterministic, a design which has many replications in some coordinates or which has points clustered on some area may not be desirable. Actually entropy-optimal designs often have many replicated coordinate values around the edge of the region (see Sherwy and Wynn). IMSE-optimal designs have a tendency that design sites are clustered together for small θ, with a few sites in the middle of the design region for large θ. These geometric phenomena usually do not appear in Lhd which is described next.

Latin-hypercube Design: Suppose that $(G_1,...,G_d)$ be the distribution functions of the independent input variables $(X_1,...,X_d)$, and x_{ij} be the i-th value of the j-th variable X_j for $i=1,...,n$ and $j=1,...,d$. Define $P=(p_{ij})$ to be an $n \times d$ matrix, where each column of P is an independent random permutation of $(1,...,n)$. Moreover let r_{ij} be $n \times d$ values of *i.i.d.* uniform [0,1] random variables independent of P.

Then the design sites x_{ij} of a Lhd (or a *random Lhd*) is defined by

$$x_{ij} = G_j^{-1}\left(\frac{1}{n}(p_{ij}-r_{ij})\right). \tag{2.9}$$

We see that $p_{i1},...,p_{id}$ determine in which "cell" a design site x_{ij} is located, and $r_{i1},...,r_{id}$ determine where the design site x_{ij} is located in the cell (see Figure 1 in Stein (1987) for an example of 2-dimensional 7-points Lhd or Figure 2(b) for an example of 2-dimensional 9-points Lhd with $r_{ij}=1/2$). When $r_{ij}=1/2$ for all i and j, we call this design a *midpoint Latin-hypercube design* (abbreviatedly, MLhd) or a fixed Lhd, which is used at the first stage algorithm in the next section.

McKay, Beckman and Conover (1979), and Stein (1987) showed that Lhd is more efficient, in estimating the expected value of the response ($\overline{Y}$) of computer experiments, than simple random sampling. Stein (1987) also found that the closer the response $Y(\underline{x})$

is to additive in the input variables, the more Lhd helps.

Lhd has some advantages that it is computationally cheap to construct, flexible for various input distributions and covers the design region well without replicated coordinate values. However, since Lhd is not related to any optimality of design, some Lhd's may be poor in estimating expected response and/or in predicting responses at the untried input sites.

Optimal Latin-hypercube Designs: Thus some disadvantages of both Lhd's and optimal designs can be overcome by selecting the best Lhd which optimizes a given criterion (entropy or IMSE): optimal Latin-hypercube design (OLhd).

When the responses $Y(\underline{x})$ are correlated to some degree and additive in the input variables, OLhd with an appropriate θ in (2.2) may be a desirable design for computer experiment, because it is still a Lhd and an optimal design. OLhd is also expected to be robust against the misspecification of prior θ, compared to the optimal designs, because OLhd is restricted to Lhd which could geometrically (by the blocking structure) prevent optimal design sites from being sensitively influenced by the selected value of θ. (Here note that the optimal design is searched for a given prior θ.)

On the other hand, OLhd loses the biggest benefit of Lhd which is ease of construction. Thus the development of a fast algorithm for finding the OLhd is an important task.

3. Algorithm for Optimal Latin-hypercube Design

Finding an optimal Lhd using standard optimizers (e.g., Newton-type nonlinear programming routines) seems difficult, because the feasible region is disconnected. Thus we use a 2-stage exchange and Newton-type algorithm. The algorithm first finds an *optimal midpoint Lhd* (abbreviatedly, OMLhd) which optimizes a given criteria among all possible MLhd's (Lhd with $r_{ij} \equiv 1/2$ in (2.9)). This procedure is equivalent to finding the optimal permutation matrix P in (2.9).

Then it is optimally released to a "non-midpoint" Lhd, that is, finding optimal r_{ij}'s in [0,1] without changing the obtained permutation matrix P of OMLhd. An exchange algorithm, but exchanging 2 points at a time subject to Latin-hypercube structure, is used for OMLhd search. Then a constrained quasi-Newton routine (Gill and Murray, 1972) is used to construct a "sub-optimal" Lhd from the above OMLhd by varying only $r_{ij} \in [0,1]$ instead of fixing $r_{ij} \equiv 1/2$. The second stage routine is easy to implement, for example by using E04JBF in NAG library.

Since there are $(n!)^{d-1}$ MLhd's for given n and d, it is expensive to find an OMLhd by an exhaustive enumeration method. By the definition of MLhd, if one point is moved to another site in the MLhd, then the point corresponding to the moved point's column or row must move to the other site to maintain Latin-hypercube structure. (Consider

2-dimensional MLhd, for example; see Figure 2.)

Thus two "elements" of the given design are exchanged simultaneously in this algorithm. That is, for fixed i_1 and i_2 ($i_1 \neq i_2$), if $x_{i_1 j_1}$ is changed to $x_{i_2 j_1}$, then $x_{i_2 j_2}$ also should be changed to $x_{i_1 j_2}$, for some j_1 and $j_2 = 1,...,d$. If we call one of such different exchanges for fixed i_1 and i_2 as a switching components in the pairs i_1 and i_2, then the MLhd is varied according to the switching components.

The algorithm first selects (several) active pairs which make the objective function value smallest by excluding that pairs from the given MLhd, then it finds the best switching components in the selected active pairs which minimizes the function value among all possible switches in the given pairs. This procedure incorporates (several) randomly selected pairs, and iterates until there is no improvement for all of the specified number of active pairs. Following is the description of the first stage algorithm for finding an OMLhd.

Step 1. For given design S and parameters, select the best active pairs which minimize the objective function for a set of n-2 design sites where the active pairs are excluded from S.

Step 2. Minimize the objective function over all possible exchanges of the coordinates within the given active pairs. Repeat Step 1-2 untill no improvement of the function from the previous one.

Step 3. After a successful stop in Step 2, select the active pairs at random which is not the same to the previous active pairs, and return to Step 2. If no improvement in Step 2, then stop.

By the exhaustive enumeration in Step 1, n evaluations of the objective function with $n-1$ points are required. In Step 2, for an active pairs, $(2^{d-1}-1)$ calculations of the function are needed. So, if we define one iteration as an execution of both Step 1 and Step 2, then the function is evaluated $n+m(2^{d-1}-1)$ times in each iteration, where m is the number of active pairs in Step 1-2 and it takes the value between 1 and 3 in the authors' computer program.

See Figure 2, 3 and 4 for examples of optimal Latin-hypercube designs obtained by the above algorithm and the authors' computer program. These designs turned out to be well spread out and frequently almost symmetric without replications. Based on more OLhd's we obtained, it is observed that OLhd's have intuitively desirable geometric properties.

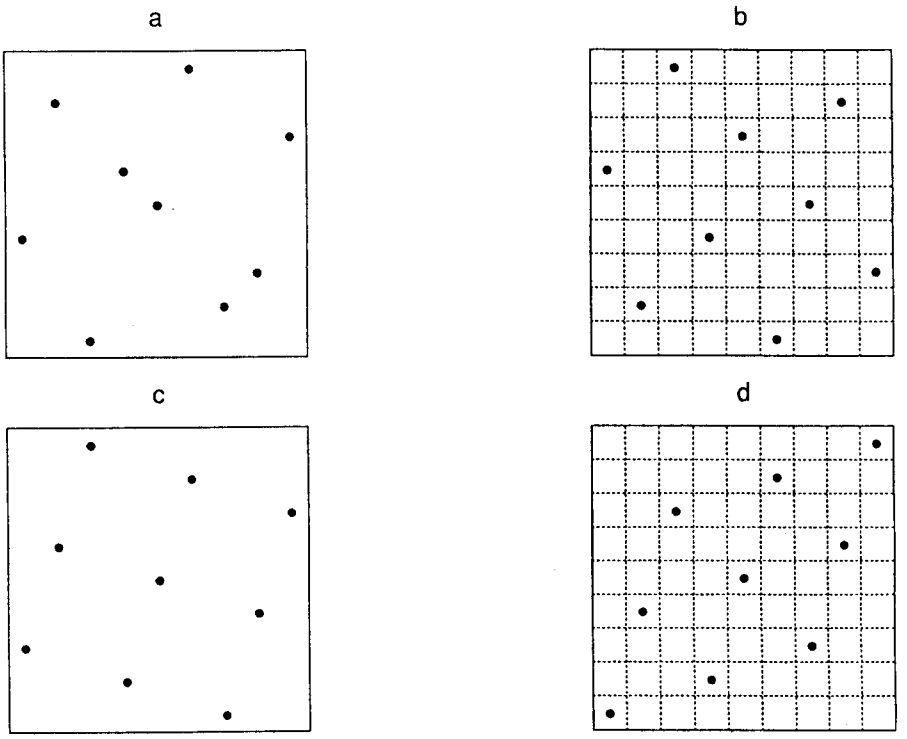

Figure 2: The IMSE-optimal midpoint Latin-hypercube designs for uniformly distributed input variables of $n=9$; (a) $\theta=1.0$ (b) $\theta=5.0$ (c) $\theta=25$ (d) $\theta=100$.

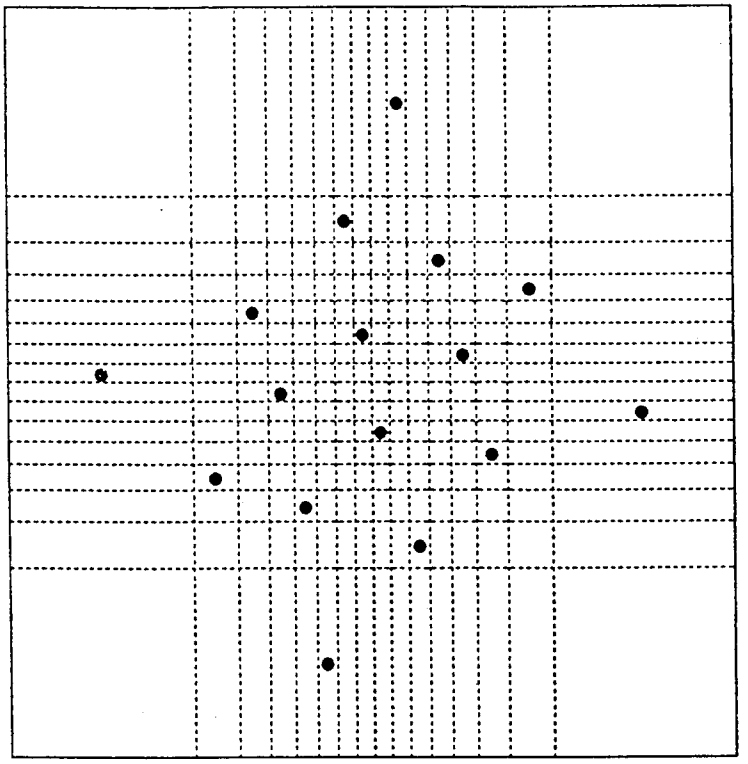

Figure 3: The entropy OMLhd for Normally distributed input variables for $n=16$ and $\theta=100$. The symmetry of the design sites is geometrically interesting.

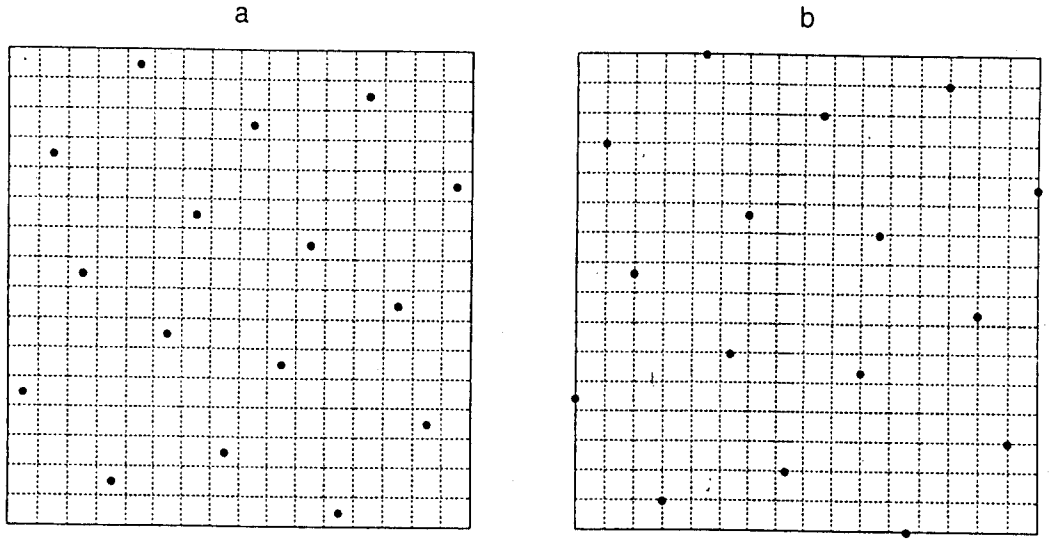

Figure 4: The entropy OMLhd (a) and its released (by a Newton-type routine) OLhd (b) for uniformly distributed input variables for $n=16$ and $\theta=26$. See the symmetries of design points.

4. Analysis of Computer Experiments

After computer data have been collected at the design sites, maximum likelihood estimators (MLE) of the model parameters are computed to build a prediction model. Since we assume $y(x)$ has a multivariate Normal distribution with mean $F\beta$ and covariance matrix $\sigma_z^2 V$, the likelihood function of y is

$$L(y;\ \theta,\beta,\sigma_z^2,\gamma_c^2,\ x) = \frac{(2\pi\sigma_z^2)^{-n/2}}{\sqrt{|V|}} \exp\left(- \frac{(y-F\beta)' V^{-1}(y-F\beta)}{2\sigma_z^2} \right).$$

When θ and γ_c^2 are specified, the MLE of σ_z^2 is given by

$$\widehat{\sigma_z^2} = \frac{1}{n}(y-F\hat{\beta})' V^{-1}(y-F\hat{\beta}),$$

where $\hat{\beta}$ is the generalized least squares estimator of β as in (2.5). Then -2 times the "concentrated" log likelihood function (except for constants) with $\hat{\beta}$ and $\widehat{\sigma_z^2}$ plugged in is

$$\lambda = n \log \widehat{\sigma_z^2} + \log |V|. \tag{4.1}$$

Here, "concentrated" refers to the fact that the likelihood has already been maximized over β and s_z^2. Since the likelihood equations do not lead to a closed form solution, a numerical optimization procedure with respect to θ is used. We used a quasi-Newton optimizer (Gill and Murray, 1972) in NAG library (E04KBF) with multiple initial values of θ because of multi-modality of the likelihood surface. Park and Cho (1994) implemented a fast algorithm for the computation of the maximum likelihood estimates.

Some combinations of β's and θ's will determine the prediction model. For example,

$Y(x)=\beta_0+Z(x)$ with a common θ, or with d different θ's,

$Y(x)=\beta_0+\beta_1 x_1+\ldots+\beta_d x_d+Z(x)$ with a common θ, or with d different θ's.

Of course, many other models are possible. The "forward" or "backward" stepwise selection procedures may be possible. See Welch et. al. (1991) for this direction. More investigation on the model selection is anticipated.

After estimating parameters, the main effects plotting (SWMW, 1989) might be used to summarize and investigate a fitted model. The estimated overall average of the response $Y(\underline{x})$ is

$$\hat{\mu}_0 = \int \hat{Y}(\underline{x}) \prod_{h=1}^{d} dx_h, \tag{4.2}$$

the estimated main effect of input factor x_i (averaged over the other factors) is the

function

$$\hat{\mu}_i(x_i) = \int \hat{Y}(\underline{x}) \prod_{h \neq i} dx_h - \hat{\mu}_0, \tag{4.3}$$

and the estimated interaction effect of x_i and x_j is the function

$$\hat{\mu}_{ij}(x_i, x_j) = \left(\int \hat{Y}(\underline{x}) \prod_{h \neq i,j} dx_h \right) - \hat{\mu}_i(x_i) - \hat{\mu}_j(x_j) - \hat{\mu}_0 \tag{4.4}$$

where $\hat{Y}(\underline{x})$ is the BLUP of $Y(\underline{x})$. See Figure 5 for an example of main-effects plot obtained for the application in the next section.

The main-effects plot is used to figure out the sensitivity of input variables to the response. Thus it is useful for identifing and selecting the important input variables to the output. We see from Figure 5 that the variables 3 and 1 are very effective on the response Y while the others are not.

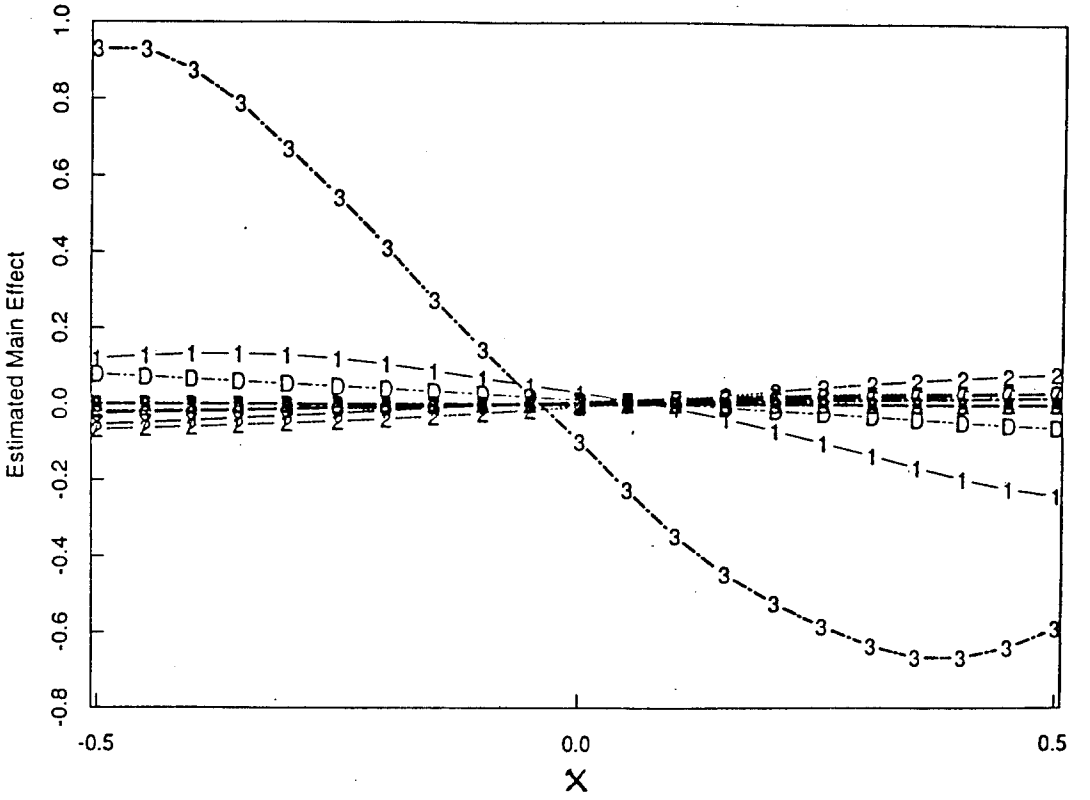

Figure 5: The estimated main effect plot of $X_1, X_2, \ldots X_{13}$ (plotting symbols 1,2,...,9,A,..,D) on current (mA). The variables X_3 and X_1 are very effective on the response Y while the others are not.

5. Application: Quality Improvement via Computer Experiments

This section deals with recent work of Jerome Sacks and William Welch (Welch and Sacks, 1991, and Welch et. al., 1990) to develope an efficient and systematic approach to quality improvemnts via computer experiments based on the spatial prediction model approach which was described in the previous sections. This section just illustrates how the methodology discussed in this paper canbe usefully applied to the quality improvements.

Many products are now routinely designed with the aid of computer models. Using relatively few runs of the computationally expensive computer model, this approach builds approximating functions (to the computer models) to be used during product-design optimization.

Integrated Circuit Design Optimization: Circuit performances generally depend on designable parameters (control factors), operating conditions (environmental noise factors), and on statistical variations of devices parameters (uncontrollable manufactruing noise factor).

Let $\underline{x}=(x_1,...,x_d)$ denote the vector of varying input parameters to the circuit simulator, all the other inputs remaining fixed. They first treat a single performance, denoted by Y, an output current. Extension to multiple performance is available (See Bernado et. al., 1992). They write $x_i=c_i+u_i$ to differentiate between the controllable and uncontrollable portions. The performance Y is, therefore, a function of $\underline{x}=\underline{c}+\underline{u}$, where $\underline{c}=(c_1,...,c_d)$ and $\underline{u}=(u_1,...,u_d)$.

They adopt Taguchi's objective of minimizing a loss, for example a measure of variability around a target performance, rather than maximizing the yield. For the circuit simulation example, the target value of the output is current 1 mA and it is the fluctuation aroud 1 mA due to $\underline{u}$. This suggests the loss structure

$$L_{MAX}(\underline{c})=\max_{\underline{u}} |Y(\underline{c}+\underline{u})-1|, \tag{5.1}$$

with the ultimate objective to minimize this loss by choice of $\underline{c}$.

A description of their general approach is: select inputs at which to run the simulator, collect the data, model $Y(\underline{c}+\underline{u})$ over the region of interest and obtain a predictor $\hat{Y}(\underline{c}+\underline{u})$. Then they minimize, over $\underline{c}$, the predicted loss

$$\widehat{L_{MAX}}(\underline{c})=\max_{\underline{u}} |\hat{Y}(\underline{c}+\underline{u})-1|. \tag{5.2}$$

The advantage is that $\hat{Y}$ is invariably far cheaper to evaluate than Y, thus making it feasible to optimize $\widehat{L_{MAX}}$, the result of which can be used to estimate the optimal $\underline{c}$. This contrasts with other strategies (e.g., Taguchi's) that attempt to directly optimize

L_{MAX} or model L_{MAX}. And sequential experimentations were used to obtain an accurate predictor in the vicinity of the optimum.

The detailed steps are as follows:

Step 1. Model the performance Y by the model (2.1).

Step 2. Design an initial experiment and run the simulator to collect the data. (Actually Latin-hypercube design was used).

Step 3. Use the data to estimate the correlation parameters in (2.2) via maximum likelihood and obtain the BLUP $\hat{Y}$.

Step 4. Decompose $\hat{Y}$ into effects due to individual parameters and joint (interaction) effects of pairs of parameters as were given by (4.2)-(4.4), and plot these main-effects (See Figure 5).

Step 5. If the predictor is not accurate enough (e.g., if a confimatory run inducates poor prediction), then determine a smaller region from which to select a next stage experimental desgin, e.g., by using the plot of Step 4, and/or contour plot (Figure 6) of the objective function. Repeat Steps 2-5.

Step 6. When the predictor is adequate, optimize the estimated loss based on $\hat{Y}$, e.g., minimize $\widehat{L_{MAX}}(\underline{c})$ in (5.1). Do a confirmatory run. Return to Step 5 if necessary.

For a detailed application of these steps to integrated circuit design example, see Bernado et. al. (1992), and Welch et. al. (1990). In summary, they followed the six steps in two stages of the first 48 Latin-hypercube design points runs and 24 more Lhd points runs of the circuit simulator, and were able to arrive at a satisfactory solution. See Figure 6 for a contour plot of $\widehat{L_{MAX}}(\underline{c})$ used in Step 5 for searching the optimal region of input variables.

The region in the box in Figure 6 was used as the second stage design and prediction regions where 24 more Lhd points were selected.

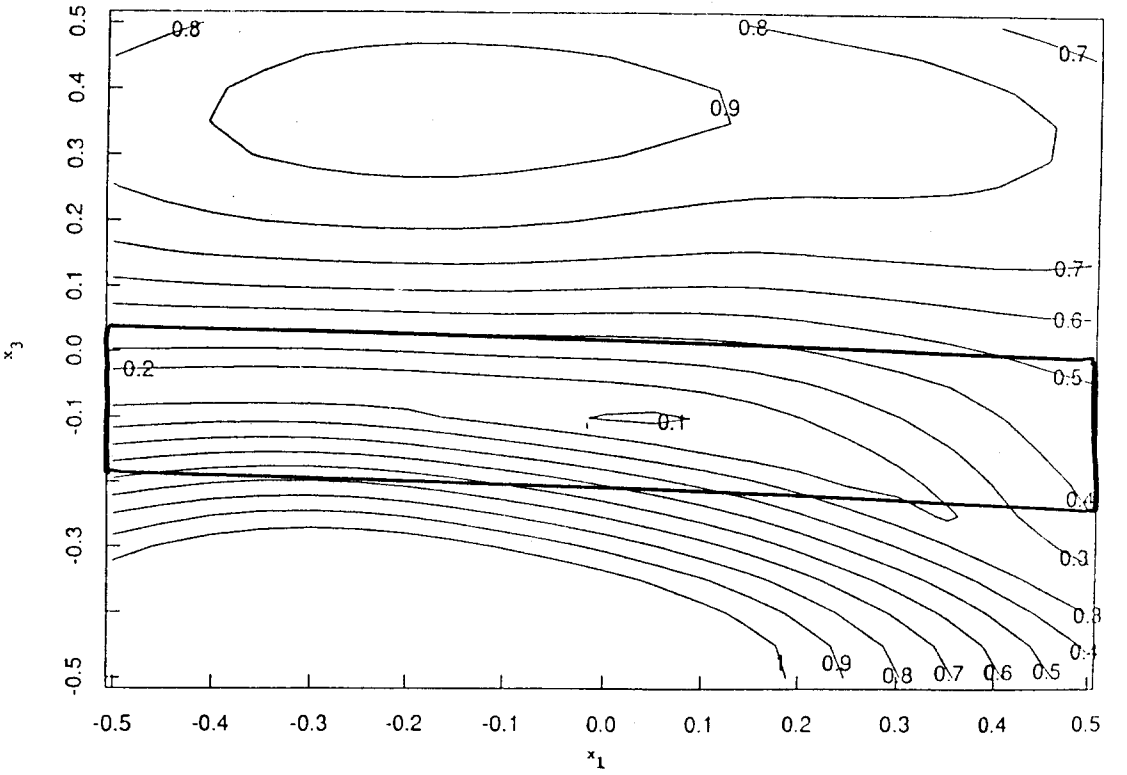

Figure 6: The contour plot of $\widehat{L_{MAX}}$, obtained from the first stage design and prediction, as a function of input variables X_1 and X_3. The region in a box was used as the second stage design and prediction regions where 24 more Lhd points were selected for the optimization of $\widehat{L_{MAX}}$.

6. Sequential (Optimal) Latin-hypercube Designs

One may re-use, as a sequential design strategy, the design sites (and its responses) of the first stage which were located in the second stage region instead of ignoring the design sites. Then we may need the strategy of ''sequential Latin-hypercube designs'' in the second stage design which seems practically useful. Moreover, the OLhd's (instead of the usual Lhd's) can be used through this sequential experimentations. Especially, in Step 4 of the previous section, one may use the estimated parameters $(\hat{\theta})$ in finding the next stage OLhd.

For updating Lhd's in a sequential manner, let a n_1 points d-dimensional Lhd is given. Suppose that we want to update n_2 more points, without invalidating the Latin-hypercube structure, to construct a n_1+n_2 point Lhd. For this purpose, first figure out the permutation integers for the given n_1 points by using the following inverse formula obtained from (2.9).

$$p_{ij}=[G_j(x_{ij})\times(n_1+n_2) + 1], \qquad i=1,...,n_1,\ j=1,...,d, \tag{6.1}$$

where the blaket $[x]$ represents the biggest integer which is less than or equal to x. Some adjustments for p_{ij} or restrictions for the number n_2 may be needed to prevent the possible replications of permutation integers in (6.1) for some j. Next, for each j, choose n_2 integers from $(1,...,n_1+n_2)$ which are not the same with the previous p_{ij} obtained in (6.1).

Then the new n_2 design sites are obtained by

$$x_{ij} = G_j^{-1}\left(\frac{p_{ij}-r_{ij}}{n_1+n_2}\right), \tag{6.2}$$

where $i=n_1+1,...,n_1+n_2$ and $j=1,...,d$. See Figure 7 for an example of sequential (optimal) Lhd's with $n_1=9$ and $n_2=9$. The cost of constructing sequential optimal Lhd's turned out to be much cheaper than that of the one-time OLhd construction.

7. Concluding Remarks

We have considered the use of Latin-hypercube designs and a spatial prediction model for design and analysis of computer experiments. As a result, optimal Latin-hypercube design turned out to have intuitively desirable geometric properties. For analysis of the simulated data, parameters of the spatial prediction model which approximates computer simulation code were estimated by the maximum likelihood method. The relationships

between inputs and outputs are visualized by the main-effects plot and contour plot.

This approach, which is viewed as a mixture of response surface methodology and spatial statistics, will help the engineer to initially identify the improtant input parameters, build approximating models, visualize relationships and proceed sequentially to a good product.

The developement of a fast algorithm for finding OLhd is anticipated especially for large n and d. Further, well implemented sequential (optimal) Latin-hypercube designs and data analysis based on the spatial model will be very useful in the practices of the computer simulation experimentations.

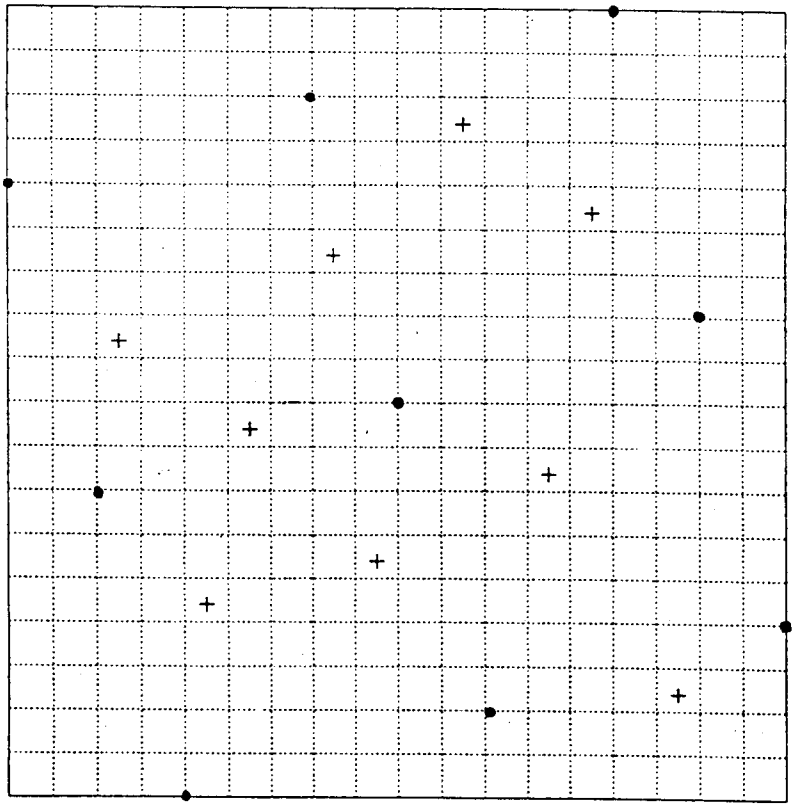

Figure 7: The entropy optimal sequential Lhd's for $n_1+n_2=18$ and $\theta=25$;

• --- the first stage 9 points (n_1) released OLhd, and + --- the second stage (updated) 9 points (n_2) OMLhd.

REFERENCES

[1] Bernado, C., Buck, R., Liu, L., Nazaret, W., Sacks, J. and Welch, W. (1992), Integrated Circuit Design Optimization using A Sequential Strategy, *IEEE Transactions on Computer-Aided-Design of Integrated Circuit and System,* Vol. 11, 362 - 371.

[2] Cressie, N. (1991). *Statistics for Spatial Data.* John Wiley, New York.

[3] Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments, *Journal of the American Statistical Association,* Vol. 86, 953-963.

[4] Gill, P. and Murray, W. (1972). Quasi-Newton Methods for Unconstrained Optimization, *Journal of The Institute of Mathematics & Its Applications,* Vol. 9, 91-108.

[5] Iman, R., and Conover, W. (1980), Small Sample Sensitivity Analysis Techniques for Computer Models, with An Application to Risk Assessment (with discussion), *Communications in Statistics,* A, Vol. 9, 1749-1874.

[6] McKay, M.D., Beckman, R.J, and 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,* Vol. 21, 239-245.

[7] Park, J. and Cho. W. (1994). An Efficient Implementation of MLE Searching Algorithm in A Gaussian Spatial Model, Technical Report, submitted.

[8] Park, J. and Park, J. (1991). Analysis of Computer Experimental Data using A Bayesian Prediction Model, *Theses Colloquium Chonnam Univ.,* Vol. 36, 153-162 (Korean).

[9] Ripley, B. (1981), *Spatial Statistics,* John Wiley, New York.

[10] Sacks, J., Schiller, S. B. and Welch, W. J. (1989), Designs for Computer Experiments, *Technometrics,* Vol. 31, 41-47.

[11] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), Design and Analysis of Computer Experiments (with discussion), *Statistical Science,* Vol. 4, 409-435.

[12] Shewry, M., and Wynn, H. (1987), Maximum Entropy Sampling, *Journal of Applied Statistics,* Vol. 14, 165-170.

[13] Stein, M. (1987), Large Sample Properties of Simulations using Latin Hypercube Sampling, *Technometrics,* Vol. 29, 143-151.

[14] Welch, W., Buck, R., Sacks, J., Wynn, H., Mitchell, T., and Morris, M. (1992), Screening, Predicting, and Computer Experiments, *Technometrics,* Vol. 34, 15-25.

[15] Welch, W. and Sacks, J. (1991), A System for Quality Improvement via Computer Experiments, *Communications in Statistics-Theory & Methods,* Vol. 20, 477 - 495.

[16] Welch, W., Yu, T., Kang, S., and Sacks, J. (1990), Computer Experiments for Quality Control by Parameter Design, *Journal of Quality Technology,* Vol. 22, 15-22.

품질 향상에 적용되는 전산 실험의 계획과 분석3)

심정욱, 박정수, 배종성4)

요약

컴퓨터 시뮬레이션 실험을 이용한 제반 연구의 효율성을 높이기 위한 통계적 실험계획법으로서 최적 실험계획법과 라틴 하이퍼큐브 계획법에 대하여 연구하여 최적 라틴하이퍼큐브 계획법을 제시하였다. 또한 전산 실험 자료의 분석을 위하여, 공간적 예측모형을 택하여 자료로 부터의 모수추정과 이 모형에 적합한 예측방법 및 최적 실험계획법등이 고려되었다. 최적 라틴하이퍼큐브 실험계획법을 구성하기 위한 2단계 (2점 교환법 및 뉴톤방법) 알고리즘과 그것에 의한 결과를 제시하였고, 나아가 축차적 (최적) 라틴 하이퍼큐브 계획법의 구축을 위한 한 방법을 제안하였다. 이와같은 접근법은 주요인 그림과 축차적인 계획 및 분석을 이용하여 집적회로 계획의 최적화 문제로 응용되어 결국 품질향상에 도움이 되도록 하는 실예를 통하여 그 실제적 적용성이 예증되었다.

3) 이 논문은 1992년도 교육부 지원 한국 학술진흥재단의 자유공모 과제 학술연구 조성비에 의하여 연구되었음. 본 논문의 요지는 1992년 6월 서울대학교에서 열린 『품질향상을 위한 실험계획법에 관한 워크숍』에서 발표되었음.

4) (500-757) 광주직할시 북구 용봉동 300 번지 전남대학교 자연과학대학 통계학과,