LA-UR-09-05818

Approved for public release; distribution is unlimited.

Title:	Batch Sequential Designs for Computer Experiments
Author(s):	Jason L. Loeppky, UBC-Okanagan Leslie M. Moore, CCS-6, LANL Brian J. Williams, CCS-6, LANL
Intended for:	Journal of Statistical Planning and Inference

Form 836 (7/06)

Batch Sequential Designs for Computer Experiments

Jason L. Loeppky
Department of Mathematics and Statistics
University of British Columbia, Okanagan
Kelowna, BC V1V 1V7, CANADA
(jason@stat.ubc.ca)

Leslie M. Moore
Statistical Sciences Group
Los Alamos National Laboratory
Los Alamos, NM, 87545, USA
(lmoore@lanl.gov)

Brian J. Williams
Statistical Sciences Group
Los Alamos National Laboratory
Los Alamos, NM, 87545, USA
(brianw@lanl.gov)

September 8, 2009

Abstract

Computer models simulating a physical process are used in many areas of science. Due to the complex nature of these codes it is often necessary to approximate the code, which is typically done using a Gaussian process. In many situations the number of code runs available to build the Guassian process approximation is limited. When the initial design is small or the underlying response surface is complicated this can lead to poor approximations of the code output. In order to improve the fit of the model, sequential design strategies must be employed. In this paper we introduce two simple distance based metrics that can be used to augment an initial design in a batch sequential manner. In addition we propose a sequential updating strategy to an orthogonal array based Latin hypercube sample. We show via various real and simulated examples that the distance metrics and the extension of the orthogonal array based Latin hypercubes work well in practice.

KEYWORDS: Computer experiment, Gaussian process, Random function, Latin hypercube sample, Maximin distance, Entropy.

1 Introduction

In many areas of science, computer models are being used in conjunction with physical experiments and even replacing them in some situations. It is usually the case that the deterministic computer model is slow to run thus requiring the use of a statistical emulator for the code output. Approximating the code via a Gaussian process is now standard practice (Sacks et al., 1989a,b; Currin et al., 1991) and has proven effective in a

large variety of situations. However, when limited data are available, achieving stability in the predictions of the emulator can be challenging. There is a reasonable amount of literature concerned with the collection of new data points to improve estimation of certain features of the space or improve the overall fit of the surface (Mitchell and Morris, 1992; Bernardo et al., 1992; Jones et al., 1998; Ranjan et al., 2008; Lam and Notz, 2008). However, most of these methods are computationally demanding when attempting to achieve a good overall prediction of the response surface. In addition some of these procedures rely on a one at a time implementation which is inefficient.

In this paper we investigate various strategies for reducing the prediction error of the fitted response surface used to emulate the code. We focus on batch sequential methods, including a method extending concepts of orthogonal array based Latin hypercube sampling (Tang, 1993; Owen, 1994) (OA-based LHS), by adding sets of design points having specified binning structure as well as dense marginal values of inputs. We focus particular attention on two distance based methods. To our knowledge distance based criteria have been used with much success (Johnson et al., 1990; Morris and Mitchell, 1995) in the selection of an initial design but have not been used as a method for selecting follow up runs. We show through various examples that these distance based methods also perform well when used to select follow up runs. An appealing feature of distance based methods is that they are computationally efficient and reasonably easy to implement in a fully batch sequential fashion.

This paper is outlined as follows. In Section 2 we introduce the Gaussian process prior and show how this can be used to model the output of a deterministic simulator. In Section 3 we review the common procedures for selecting follow up runs and introduce two distance based methods and an extension of OA-based LHS for the selection of new batches of design points. In Section 4 we discuss the results from implementing the procedures on a wide variety of computer models. Finally, we make some concluding remarks and discuss future work in Section 5.

2 Gaussian Process Model

Following the path taken in the literature (Sacks et al., 1989a,b; Currin et al., 1991; O'Hagan, 1992), we place a Gaussian process (GP) prior on the possible output functions from the computer code. That is, if we let $y(\mathbf{x})$ be the output of the code for a given vector valued input $\mathbf{x} = (x_1, \ldots, x_d)$ and let $Y(\mathbf{x})$ denote a random function model for $y(\mathbf{x})$, the GP model specifies

$$Y(\mathbf{x}) = \mu + Z(\mathbf{x})$$

where μ is a mean parameter and $Z(\mathbf{x})$ is a Gaussian stochastic process with mean zero and constant variance σ^2. Typically, we take $\mu = 0$ and model $y(\mathbf{x}) - \bar{y}$ as a Gaussian process. In what follows it is assumed that $\mu = 0$. The key aspect of the GP model is the correlation structure. Following Sacks et al. (1989a) a Gaussian correlation function is used. At two input vectors, $\mathbf{x}$ and $\mathbf{x}'$, we take the correlation between $Y(\mathbf{x})$ and $Y(\mathbf{x}')$ as

$$R(\mathbf{x}, \mathbf{x}') = \exp\left(- \sum_{i=1}^{d} \theta_j (x_i - x_i')^2 \right).$$

Each correlation range parameter θ_j specifies the degree to which model output is correlated as the j-th input varies; smaller values of θ_j correspond to longer correlation lengths.

Given an n run initial design X_0 for input vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ in $[0,1]^d$, the data $\mathbf{y} = (y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)}))^T$ are collected by running the computer model. The predictor $\hat{Y}(\mathbf{x})$ of $Y(\mathbf{x})$ is the posterior mean of $Y(\mathbf{x})$ given the data and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$:

$$\hat{Y}(\mathbf{x}) = E\left(Y(\mathbf{x}) | \mathbf{y}, \boldsymbol{\theta}\right) = \mathbf{r}^T(\mathbf{x}) \mathbf{R}^{-1} \mathbf{y}\,, \tag{1}$$

where $\mathbf{r}(\mathbf{x}) = (R(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, R(\mathbf{x}, \mathbf{x}^{(n)}))^T$ is an $n \times 1$ vector, $\mathbf{R}$ is an $n \times n$ matrix with element i, j given by $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, and the mean square error (MSE) of $\hat{Y}(\mathbf{x})$ is given by

$$\mathrm{MSE}(\hat{Y}(\mathbf{x})) = E\left(\hat{Y}(\mathbf{x}) - Y(\mathbf{x})\right)^2 = \sigma^2 \left(1 - \mathbf{r}^T(\mathbf{x}) \mathbf{R}^{-1} \mathbf{r}(\mathbf{x})\right). \tag{2}$$

In practice the parameters σ^2 and $\boldsymbol{\theta}$ must be estimated from the initial code runs. This can be done by providing prior distributions on the hyper-parameters of the process and

performing a Bayesian analysis as outlined in Higdon et al. (2004) and Higdon et al. (2008). Alternatively, one can follow Currin et al. (1991) and adopt an empirical Bayes approach and estimate the parameters by maximizing the likelihood. See Welch et al. (1992) for a full specification of the likelihood. For the purposes of this paper and the ease of implementation in a sequential procedure we use the empirical Bayes approach. However, it is important to note that once all the data is collected and one wishes to make predictions of the process a fully Bayesian formulation is preferable since it takes account of the uncertainty in parameter estimation and provides more complete assessments of prediction uncertainty.

Given the above model specification, the outstanding issues we address in this paper are focused on strategies for improving the approximation of the code if the initial design X_0 is found to give inadequate predictions of the underlying response surface. Initially we select an OA-based or maximin Latin hypercube design (McKay et al., 1979; Morris and Mitchell, 1995; Tang, 1993) since they have desirable space filling properties and have proven to work well in a variety of circumstances. In selecting an initial design it is desirable to choose a run size that will allow for an adequate initial approximation of the surface. Ranjan et al. (2008) noted that initial surface estimates were adequate for the performance of their sequential design algorithm if an initial design consumed roughly 25% - 35% of a fixed budget of runs. Loeppky et al. (2008) provides a justification for using the often quoted rule of thumb of selecting a sample size that is 10 times the number of dimensions. Although $n = 10$ should provide a reasonable fit, how to select additional points when 10 or fewer runs are insufficient is still an open issue. In what follows we will take run sizes smaller than $n = 10$ for illustrative purposes but our experience is that 10 , or at a minimum 10 times the number of assumed active inputs, is recommended.

3 Criteria for Emulator Maturity

In this section we address the question of how one might choose a set of follow up runs to improve the overall fit of the emulator. That is, how do we achieve a certain level of emulator maturity. There has been a reasonable amount of work in the literature suggesting various methods for improving the fit of the model. Mitchell and Morris (1992) and Bernardo et al. (1992) implemented sequential design strategies for a computer model. Mitchell and Morris (1992) began by fitting a smaller run design than permitted by the final budget and added sets of 10 points selected from a candidate set and chosen to maximize the entropy (Shewry and Wynn, 1987) of the overall design. Whereas Bernardo et al. (1992) added additional points by narrowing the space and adding extra points using a Latin hypercube design, Sacks and Schiller (1988) and Sacks et al. (1989b) investigated the use of integrated MSE as a method for selecting design points. Sacks and Schiller (1988) also considered the strategy of adding points to minimize the maximum MSE. More recently Aslett et al. (1998) considered batch sequential designs for a problem in circuit optimization. In this context the strategy employed was to narrow the input region by selecting areas of the space that have a high amount of activity, followed by running Latin hypercube designs in these smaller regions of space.

In recent years the use of expected improvement criteria has gained much popularity. Jones et al. (1998) considered an expected improvement criterion for global optimization of the response surface, implemented both in a fully sequential (one at a time) and batch sequential manner. Recently Ranjan et al. (2008) considered the use of the expected improvement to estimate the contour of a computer model. Finally, Lam and Notz (2008) considered a modification of the expected improvement of Jones et al. (1998) that may be used for achieving good global fit (emulator maturity) of the model.

Recently, the addition of new points has been done in a fully sequential manner. For the most part this is due to the complicated nature of the criteria and the fact that methods relying on the data $\mathbf{y}$ are not readily extendable to a batch sequential strategy. However, adding points one at a time can often lead to suboptimal placements

of points. In addition, it is common for multiple processors to be used simultaneously to produce sets of runs from the code. In such cases, one at a time sequential strategies are impractical. In a situation where the code could take upwards of 10 hours to run, a fully sequential design strategy is simply not possible due to the inordinate amount of computing time that would be required to produce an emulator that achieves a high level of accuracy across the entire input domain. In any practical situation it is imperative to operate in a batch sequential fashion and find sets of points.

Consider an initial design X_0 which will be augmented to a new design $X_1 = (X_0^T, X_b^T)^T$ where X_b is a $m \times d$ matrix. The rows of X_b are elements of a set of candidate points $\mathcal{F}$. In what follows we assume that $\mathcal{F} \equiv [0,1]^d$. If the updates are fully sequential $m = 1$, and otherwise $m > 1$ and the updates are batch sequential. With the exception of the expected improvement for global fit algorithm (Lam and Notz, 2008), it is assumed that $m > 1$ is used. Design augmentation is implemented with the following batch sequential algorithm.

1. Estimate the GP covariance parameters σ^2 and $\boldsymbol{\theta}$ using computer model runs from the initial design X_0.

2. Set $X_1 = (X_0^T, X_b^T)^T$ and obtain X_b by optimizing a design criterion with respect to the proposed m additional runs. The design criteria investigated in the examples of Section 4 are explained below. Continuous optimizations are carried out using a modified Federov exchange (Fedorov, 1972).

3. Collect computer model runs from X_b and re-estimate the GP covariance parameters σ^2 and $\boldsymbol{\theta}$ using the entire set of runs from the augmented design X_1.

4. Set X_0 to the augmented design X_1 and repeat steps (2) and (3) until termination. Relevant stopping criteria include consumption of an allotted budget of total runs or failure to observe significant improvement in the design criterion value or in predictive performance of the emulator.

We now introduce the statistically based criteria evaluated for design augmentation,

followed by two distance based methods (Johnson et al., 1990) motivated by the statistical criteria.

Lam and Notz (2008) considered choosing additional points that maximize the expected improvement

$$E(I(\mathbf{x})) = E\left((Y(\mathbf{x}) - y(\mathbf{x}^*))^2\right) = MSE\left(\hat{Y}(\mathbf{x})\right) + \left(\hat{Y}(\mathbf{x}) - y(\mathbf{x}^*)\right)^2$$

where $MSE\left(\hat{Y}(\mathbf{x})\right)$ is given in (2) and $\mathbf{x}^*$ is a point in the existing design X_0 that is closest to $\mathbf{x}$. The first part of the equation can be updated in a batch sequential fashion. It is not obvious how to update the second half of the formula since one may need the new value of $y(\mathbf{x}^*)$ where $\mathbf{x}^*$ may not be a point in the existing design. One possibility would be to use $\hat{Y}$ in place of y on X_b with a modified MSE (Williams et al. (2009)); however, we have found that this does not work well in practice.

Sacks and Schiller (1988) considered a minimax strategy where one selects a design X that minimizes the maximum MSE. In the context of sequential design strategy one has

$$\min_{X_b \subset \mathcal{F}} \max_{\mathbf{x} \in [0,1]^d} MSE\left(\hat{Y}(\mathbf{x})\right)$$

where $MSE\left(\hat{Y}(\mathbf{x})\right)$ is given in (2) but $\mathbf{R}$ is computed for the entire design $X_1 = (X_0^T, X_b^T)^T$. Notice that $\max_{\mathbf{x} \in [0,1]^d} MSE\left(\hat{Y}(\mathbf{x})\right)$ involves numerically finding the point $\mathbf{x}$ in $[0,1]^d$ that results in the largest possible MSE. When the set of candidate points is continuous the above criterion will be computationally intensive as the MSE criterion involves a numerical optimization within the design optimization (Sacks and Schiller, 1988).

Sacks and Schiller (1988) and Sacks et al. (1989b) considered choosing design points that minimize the integrated MSE (IMSE). In a sequential design strategy one would choose the design X_b that results in

$$\min_{X_b \subset \mathcal{F}} \int_{[0,1]^d} MSE\left(\hat{Y}(\mathbf{x})\right) d\mathbf{x}$$

where $MSE\left(\hat{Y}(\mathbf{x})\right)$ is given in (2) but $\mathbf{R}$ is computed for the entire design $X_1 = (X_0^T, X_b^T)^T$. Analytic expressions for the IMSE can be found that greatly improve the

computational time compared to the max MSE. However, the IMSE is still computationally intensive and can also be prone to fitting problems due to instability in computing the inverse of the correlation matrix; in such cases we add Toby Mitchell's "wee small bit" to the diagonal of the covariance matrix. In practice we have found this works well.

Shewry and Wynn (1987) considered choosing designs that maximize the entropy. The criterion of maximum entropy characterizes the amount of information in an experiment introduced by Blackwell (1951) and further expanded by Lindley (1956). In the context of Gaussian processes Shewry and Wynn (1987) showed that maximum entropy is equivalent to maximizing the determinant of the correlation matrix $\mathbf{R}$ when there is no regression term in the model. Partitioning

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{pmatrix}$$

where $\mathbf{R}_{11}$ is the block of the correlation matrix corresponding to the initial design X_0, $\mathbf{R}_{21} = \mathbf{R}_{12}^T$ is the correlation between X_0 and X_b and $\mathbf{R}_{22}$ is the correlation matrix corresponding to X_b. One wishes to choose a design X_b that maximizes the entropy. Specifically find X_b such that,

$$\max_{X_b \subset \mathcal{F}} \det(\mathbf{R}_{11}) \det(\mathbf{R}_{12} - \mathbf{R}_{12}\mathbf{R}_{11}^{-1}\mathbf{R}_{21}) \equiv \max_{X_b \subset \mathcal{F}} \det(\mathbf{R}_{12} - \mathbf{R}_{12}\mathbf{R}_{11}^{-1}\mathbf{R}_{21}).$$

In a design-optimization algorithm this criterion is reasonably efficient in terms of computational time.

As shown in Loeppky et al. (2008) one can place an upper bound on the MSE by considering the point $\mathbf{x}^{(i)}$ that is closest (in terms of correlation) to a new point $\mathbf{x}$. That is,

$$MSE\left(\hat{Y}(\mathbf{x})\right) \leq 1 - \exp\left(-2\sum_{j=1}^{d} \theta_j (x_j - x_j^{(i)})^2\right).$$

Choosing to work with the upper bound of the MSE it is easy to show that the minimax MSE criterion translates to a maximin weighted (by the GP correlation parameters) distance criterion. Alternatively, Johnson et al. (1990) show that minimizing $1 - \det(\mathbf{R})$ results in maximizing the minimum weighted distance. Specifically, new design points

are selected using the maximin weighted distance criterion (Johnson et al., 1990),

$$\max_{X_b \subset \mathcal{F}} \min_{\mathbf{x}, \mathbf{x}' \in X_1} \sqrt{\sum_{j=1}^{d} \theta_j (x_j - x'_j)^2}\,.$$

There are a number of appealing features of this criterion. First, it is based on both the maximum entropy criterion and the maximum mean square error criterion. Secondly, it readily incorporates the information obtained from the data by using the correlation parameters. Lastly, from a practical standpoint it is much more computationally tractable than the other design criteria discussed above as it does not involve the inversion of the correlation matrix, which is very costly in terms of computational time.

Following along the lines of the maximin weighted distance criterion one could simply choose points using the maximin Euclidean distance criterion,

$$\max_{X_b \subset \mathcal{F}} \min_{\mathbf{x}, \mathbf{x}' \in X_1} \sqrt{\sum_{j=1}^{d} (x_j - x'_j)^2}\,.$$

When all the correlation parameters are equal, this is equivalent to the maximin weighted distance criterion. If any of the correlation parameters are estimated to be zero it is initially unclear if this will adversely affect the maximin weighted distance criterion and it may be desirable to use straight maximin distance. However, we have found in practice that correlation parameters equal to zero do not adversely affect the current implementation of the maximin weighted distance criterion. In fact, the maximin weighted distance criterion performs extremely well in a situation where there is some degree of sparsity (ie. unequal values or zero values for some of the correlation parameters).

Additionally, for the maximin distance criterion, we incorporate a binning requirement for a batch of sequential points to be added in a spirit similar to OA-based Latin hypercube design (Tang, 1993; Owen, 1994) referred to as sequential bin-based LHS. Whether or not the initial design is an OA-based LHS, the values of the inputs can be 'binned' into a specified number of groups such that the 'binned' design is a subset of a factorial design. In the case of an OA-based LHS, the initial 'binned' design is actually an orthogonal array, possibly of strength higher than 2. Here a sequential batch

of design points is required to 'bin' in such a way as to complement the initial 'binned' design. Further, in the spirit of Latin hypercube sampling (McKay et al., 1979), the sequential batch of points is also selected to fill in values in unrepresented strata of the marginal inputs defined by the number of runs. Specifically, maximin distance is used to select a current best random assignment of strata to bin levels. The resulting sequential bin-based LHS designs have specified 'binning' structure and dense values of marginal inputs. This extension of OA-based LHS is used in the following sections and illustrated more fully in the three dimensional example. In the following section we compare the performance of the design criteria outlined in this section.

4 Results

4.1 Two Dimensional Examples

In this section we use a two-dimensional Branin function to compare batch sequential additions and the fully sequential strategy. The Branin function is given by

$$y = \left(x_2 - \frac{5.1}{4^{\ 2}}x_1^2 + 5\frac{x_1}{} - 6\right)^2 + 10\left(1 - \frac{1}{8}\right)\cos(x_1) + 10$$

where $x_1 \in [-5, 10]$ and $x_2 \in [0, 15]$. In order to fit the function we take an initial maximin LHS with $n_0 = 5$ runs and augment this design up to a final run size of $n = 21$. This is done using a fully sequential strategy where a single new point is added and the GP fit of the function is updated after each new run. We also make batch sequential updates where two batches of 8 new runs are selected using a modified Fedorov exchange (Fedorov, 1972). The GP model is updated after each set of 8 runs has been performed.

In order to compare the two methods we take a random LHS of m points in $[0, 1]^2$ and compute the root mean square error and the absolute value of the maximum prediction error computed for these holdout data. Specifically,

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y(\mathbf{x}^{(i)}) - \hat{y}(\mathbf{x}^{(i)}))^2}$$

and

$$\text{Maximum error} = \max_{\mathbf{x}^{(i)} \in \mathcal{H}} |y(\mathbf{x}^{(i)}) - \hat{y}(\mathbf{x}^{(i)})|$$

where $\mathcal{H}$ denotes the holdout data, $y(\cdot)$ the calculated output, and $\hat{y}(\cdot)$ the predicted output computed as in (1).

The results in Table 1 show the RMSEs and the maximum error for the maximin distance and weighted distance criteria, maximum entropy and IMSE for the batch versions and the one at a time updates. In all cases the batch algorithms outperform the fully sequential versions in terms of maximum error and RMSE.

Table 1: Branin Function: Comparison of fits for batch sequential and fully sequential implementations of the criteria.

	Batch Sequential		Fully Sequential	
	RMSE	Maximum Error	RMSE	Maximum Error
Distance	0.0375	0.1728	0.0530	0.2011
Weighted Distance	0.0462	0.2138	0.0506	0.2406
Entropy	0.0335	0.0951	0.0526	0.2049
IMSE	0.0670	0.9217	0.3132	3.2109

The plots in Figure 1 compare the design points selected using the batch sequential strategy (blue stars) to the fully sequential strategies (red squares). The points are overlaid on the contour lines of the true function. The batch sequential versions tend to spread the points more evenly compared to the one at a time versions. This is especially true for IMSE where the one at a time implementation tends to cluster points close together, which explains the poor performance of this method. In the following sections we investigate more fully the batch sequential versions of the design criteria for problems in 3, 4 and 11 dimensions.

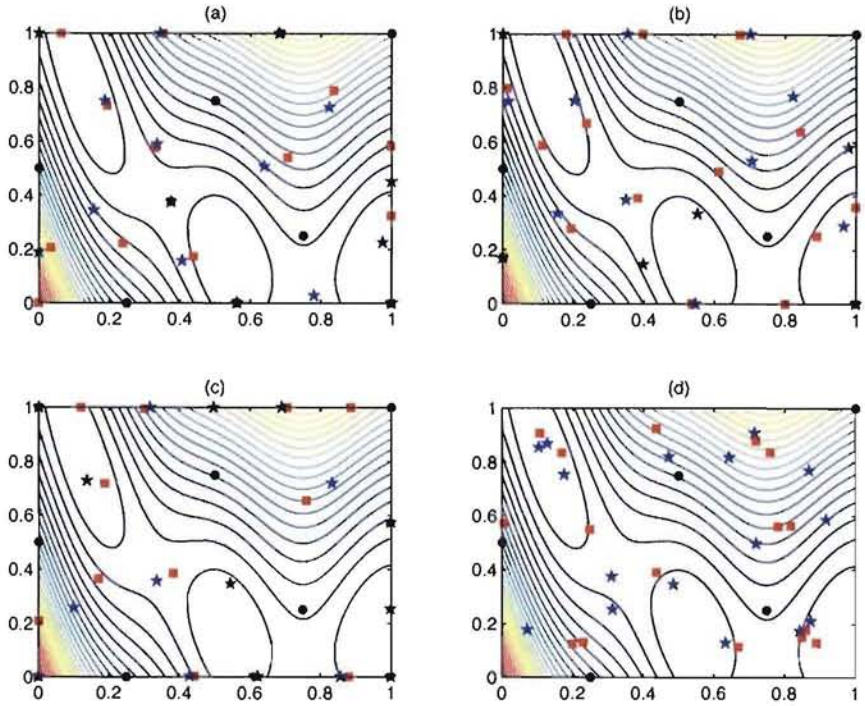

Figure 1: Branin Function: Contour plot of the true function and the design used to fit the function. Black circles are the initial 5 run maximin LHS, blue stars are the points added in batches and red squares are the one at a time sequential updates. (a) shows the maximin distance criterion, (b) maximin weighted distance, (c) maximum entropy and (d) integrated mean square error.

4.2 Three Dimensional Examples

In order to test the performance of the design criteria under an ideal situation we generate data from a three dimensional Gaussian process on a 33^3 grid of points in $[0,1]^3$. By simulating data from the prior process used to model the data we do not have to be concerned about the inadequacies of using an incorrect model to fit the data. That is, data from a computer model is not generated from a GP prior and inevitably there is some error introduced by fitting the GP model to the data. However, in practice this is usually not a concern since the GP model often provides an adequate fit.

In comparing the design criteria we also consider the naive approach of using a Sobol sequence (Sobol, 1979), which is easily generated and augmented to produce batches of additional runs independent of the design criteria studied here. We also consider a fixed maximin Latin hypercube design (Morris and Mitchell, 1995) with the full final budget of runs. With the exception of the expected improvement algorithm which is implemented in a fully sequential fashion, the other criteria are implemented in a batch sequential manner selecting batches of size 8. With the exception of the sequential bin-based LHS, the batches of size 8 are selected using a continuous optimizer and an exchange algorithm.

A final budget of 64 runs is used to fit the model, representing about $20d = 60$ runs and accommodating use of a full 4^3 underlying factorial structure for the fixed design case and sequential bin-based LHS. The fixed design is an OA-based LHS with 64 runs. The sequential strategy starts with an initial design, X_0, that is an OA-based LHS with 16 runs. This design was used as the initial design for each criterion in turn. The underlying bin structure for the initial design, X_0, is a strength 2 orthogonal array for three 4-level factors, B_0, defined by the following modulus 2 arithmetic defining relation for 6 two-level factors identified as x_1, x_2, x_3, x_4, x_5, and x_6:

$$0 = (x_1 + x_2 + x_3 - x_5) = (x_1 + x_3 + x_4 - x_6) = (x_2 + x_4 - x_5 - x_6)$$

and associating two 2-level factors with a single 4-level factor by contractive replacement (Hedayat et al. (1999), pages 204, 272). Here (x_1, x_2), (x_3, x_4), and (x_5, x_6) are associated with three 4-level factors and we associate the integers 0 and 1 with 2-level factors and 0, 1, 2, 3 with 4-level factors.

For the sequential bin-based LHS strategy, the batches of 8 runs all have underlying factorial bin structure that complement the initial bin structure B_0 and sequentially augment it so that the aggregate bin design with 6 sets of 8 runs and the initial set of 16 runs is the full 64 runs of the 4^3 factorial design. The sequential bins underlying the sequential bin-based LHS designs are defined by the following sequence of defining relations between six 2-level factors and, again, associating two 2-level factors with a single 4-level factor. B_1 and B_2 together are the 16 runs defined by folding over the

value of x_6 in the runs as defined in B_0 and are the half of these runs defined by the additional modulus 2 contrasts $(x_1+x_2-x_4)=0$ and $(x_1+x_2-(x_4+1))=0$ respectively. Specifically B_1 is defined such that:

$$\begin{aligned} 0 &= (x_1+x_2-x_4)=(x_1+x_2+x_3-x_5)=(x_3-x_4-x_5) \\ &= (x_1+x_3+x_4-(x_6+1))=(x_2+x_3-(x_6+1))=(x_2+x_4-x_5-(x_6+1)) \\ &= (x_1-x_5-(x_6+1)) \end{aligned}$$

and B_2 is defined such that:

$$\begin{aligned} 0 &= (x_1+x_2-(x_4+1))=(x_1+x_2+x_3-x_5)=(x_3-(x_4+1)-x_5) \\ &= (x_1+x_3+x_4-(x_6+1))=(x_2+x_3-x_6)=(x_2+x_4-x_5-(x_6+1)) \\ &= (x_1-x_5-x_6) \end{aligned}$$

again with (x_1,x_2), (x_3,x_4), and (x_5,x_6) associated with three 4-level factors. At this stage, the aggregate bin design $(B_0^T,B_1^T,B_2^T)^T$ is 32 runs defined by the full half fraction defined by $(x_1+x_2+x_3-x_5)=0$. The next four bin sets are the 32 runs of the other half fraction defined by $(x_1+x_2+x_3-(x_5+1))$ so that B_3 is defined by:

$$\begin{aligned} 0 &= (x_1+x_2-x_4)=(x_1+x_2+x_3-(x_5+1))=(x_3-x_4-(x_5+1)) \\ &= (x_1+x_3+x_4-x_6)=(x_2+x_3-x_6)=(x_2+x_4-(x_5+1)-x_6) \\ &= (x_1-(x_5+1)-x_6), \end{aligned}$$

B_4 is defined by:

$$\begin{aligned} 0 &= (x_1+x_2-(x_4+1))=(x_1+x_2+x_3-(x_5+1))=(x_3-(x_4+1)-(x_5+1)) \\ &= (x_1+x_3+x_4-x_6)=(x_2+x_3-(x_6+1))=(x_2+x_4-(x_5+1)-x_6) \\ &= (x_1-x_5-x_6), \end{aligned}$$

B_5 is defined by:

$$\begin{aligned} 0 &= (x_1+x_2-x_4)=(x_1+x_2+x_3-(x_5+1))=(x_3-x_4-(x_5+1)) \\ &= (x_1+x_3+x_4-(x_6+1))=(x_2+x_3-(x_6+1))=(x_2+x_4-(x_5+1)-x_6) \\ &= (x_1-(x_5+1)-(x_6+1)), \end{aligned}$$

and B_6 is defined by:

$$\begin{aligned} 0 &= (x_1 + x_2 - (x_4 + 1)) = (x_1 + x_2 + x_3 - (x_5 + 1)) = (x_3 - (x_4 + 1) - (x_5 + 1)) \\ &= (x_1 + x_3 + x_4 - (x_6 + 1)) = (x_2 + x_3 - x_6) = (x_2 + x_4 - (x_5 + 1) - (x_6 + 1)) \\ &= (x_1 - x_5 - x_6 + 1). \end{aligned}$$

The sequences of 'bins' constructed in this way are regular factorial designs that are translates of 8 runs defined by the contrasts $(x_1 + x_2 - x_4)$, $(x_1 + x_2 + x_3 - x_5)$, and $(x_1 + x_3 - x_6)$. This construction depends on defining bins as regular fractions such that the initial 16 runs, consisting of two of the translates with contractive replacement, is a maximal strength 2 orthogonal array for the three 4-level factors. Distance values could be used as well but are not very interesting for strength 2 orthogonal arrays with only three 4-level factors, although in the following 4-d and 11-d examples, distance values are used to identify better regular fractional factorial bin sequences. As additional batches of 8 runs are included, the aggregate designs are reasonable fractional factorial designs, and particularly the full budget design bins to the full 4^3 factorial. Optimal sequences need not be restricted to standard regular fractions of factorial designs which are used here to illustrate concepts. Other criteria such as minimum aberration or near orthogonality (including balance) will be investigated in future work.

Once a sequential bin structure, B_0, B_1, ..., B_6, is identified, LHS concepts are applied. Just as the initial bin design, B_0 forms the substructure for the LHS X_0, generated to optimize the maximin distance criterion, the bins are also used to construct augmenting sets of runs that yield as nearly as possible aggregate designs that are LHS incorporating near maximin distance at each batch stage. Specifically, at each stage the augmented design is constructed as follows:

1. In each input dimension, divide the bins into equal-width strata, where the number of strata corresponding to each bin is equal to the number of runs in the current design that project into that bin, plus the number of runs assigned to that bin in the proposed augmentation.

2. In each input dimension, assign proposed new runs to strata not currently containing any runs from the current design, and optimize this assignment with respect to the maximin distance criterion applied to the augmented design.

Augmented designs constructed in this fashion will possess space-filling projection properties inherited from the bin structure, while maintaining dense marginal projections. This sequential binning strategy is thus similar in principle to OA-based LHS (Tang, 1993), although at any given stage the current design need not be a LHS nor does the current bin structure need to be balanced.

We illustrate the sequential binning strategy with a simple example. Suppose we have three inputs and the following initial designs:

B_0			X_0		
0	0	0	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
0	1	1	$\frac{3}{8}$	$\frac{7}{8}$	$\frac{5}{8}$
1	0	1	$\frac{5}{8}$	$\frac{1}{8}$	$\frac{7}{8}$
1	1	0	$\frac{7}{8}$	$\frac{5}{8}$	$\frac{1}{8}$

The bins in each dimension are $_0 = [0, \frac{1}{2}]$ and $_1 = (\frac{1}{2}, 1]$, reflecting the fact that each factor in B_0 has two levels. Since B_0 is a balanced design with four runs, each bin is divided into two strata: $_0 = [0, \frac{1}{4}] \cup (\frac{1}{4}, \frac{1}{2}]$ and $_1 = (\frac{1}{2}, \frac{3}{4}] \cup (\frac{3}{4}, 1]$. Each column of X_0 is constructed in turn by assigning the two runs at level 0 in the corresponding column of B_0 to the two strata of $_0$, with an analogous procedure for the two runs at level 1. The maximin distance criterion is used to select an optimal design from among the many possible assignments of levels to strata. Suppose now that three runs are augmented to X_0 based on the bin structure B_1:

B_1			X_b		
0	0	1	$\frac{1}{4}$	$\frac{3}{16}$	$\frac{15}{16}$
1	0	0	$\frac{11}{16}$	$\frac{7}{16}$	$\frac{1}{4}$
1	1	1	$\frac{15}{16}$	$\frac{3}{4}$	$\frac{11}{16}$

In constructing the first column of X_b, we note that the first column of $[B_0^T, B_1^T]^T$ has three runs at level 0 and four runs at level 1. The two bins are thus divided into strata as follows: $_0 = [0, \frac{1}{6}] \cup (\frac{1}{6}, \frac{1}{3}] \cup (\frac{1}{3}, \frac{1}{2}]$ and $_1 = (\frac{1}{2}, \frac{5}{8}] \cup (\frac{5}{8}, \frac{3}{4}] \cup (\frac{3}{4}, \frac{7}{8}] \cup (\frac{7}{8}, 1]$. Projecting the first column of X_0 into these strata, we find that the single run at level 0 in B_1 must be assigned to the stratum $(\frac{1}{6}, \frac{1}{3}]$, and the two runs at level 1 in B_1 must be distributed across the strata $(\frac{5}{8}, \frac{3}{4}]$ and $(\frac{7}{8}, 1]$. An analogous procedure is applied to constructing the remaining columns of X_b, noting that the partitioning of $_0$ and $_1$ into strata is column dependent when the bin structure is unbalanced. As before, the maximin distance criterion applied to the augmented design $X_1 = [X_0^T, X_b^T]^T$ is used to select an optimal X_b. Figure 2 shows the designs X_0 and X_b projected into the first two dimensions.

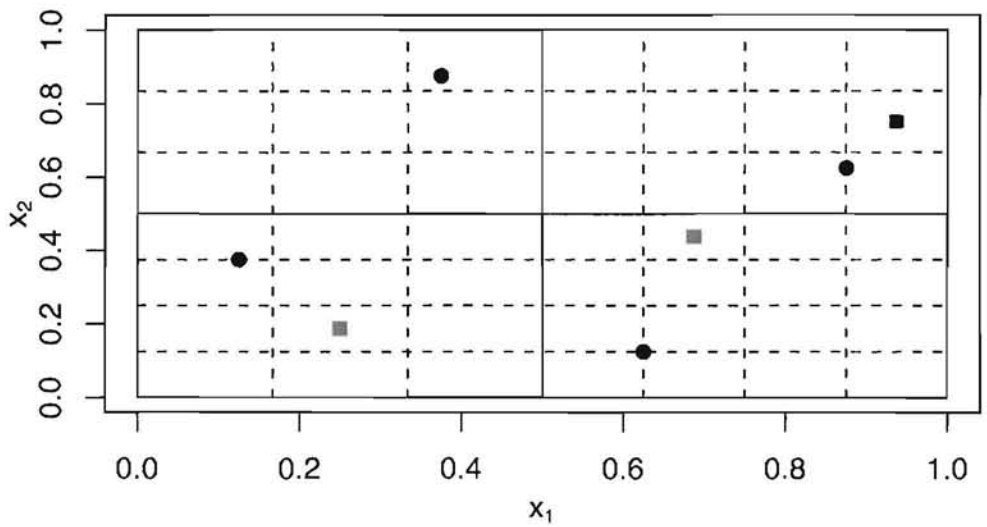

Figure 2: Illustration of sequential binning strategy with an initial four run design in three inputs augmented by a three run design. Black circles indicate the initial design X_0 and red squares indicate the augmentation X_b. Strata corresponding to the augmented design X_1 are indicated by dashed lines.

In order to test the performance of the methods over various situations 25 random realizations of the GP model are used. The overall quality of the fit is judged using both the maximum error and the RMSE computed on the 33^3 grid of points. In all cases the

values of $\sigma^2 = 1$ and $\mu = 0$ and we change the values of $\boldsymbol{\theta}$ to represent various scenarios.

Example 1: Fix $\boldsymbol{\theta} = (5, 5, 5)$ which represents a difficult problem. The size of the correlation parameters will make the surface rather bumpy and this represents an uncommon situation where all of the inputs are extremely active. In practice, we find that there is usually some degree of sparsity and only a few of the factors are important and the remaining factors have little to no impact on the response y. The plot in Figure 3 shows the maximum error and RMSE for each of the criteria.

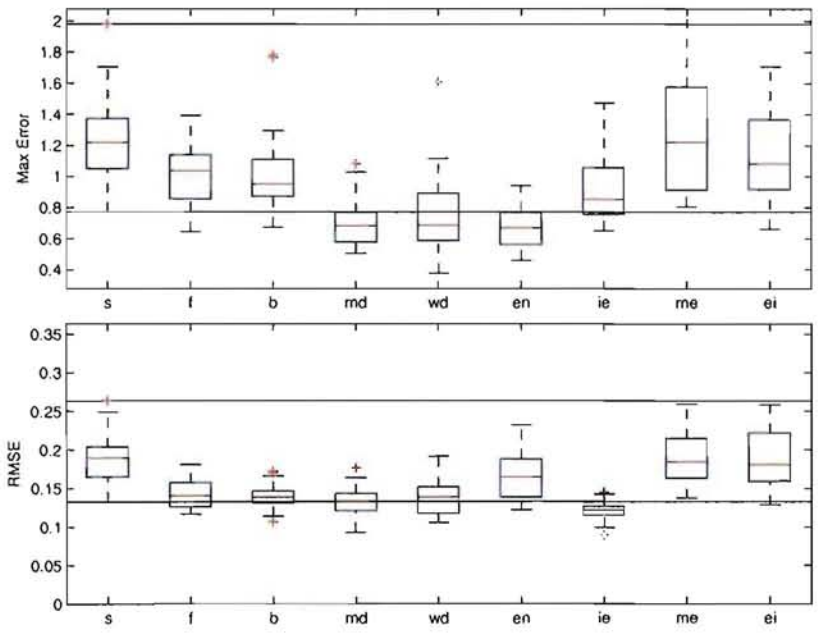

Figure 3: Generated data $\boldsymbol{\theta} = (5, 5, 5)$: Boxplots of maximum error and RMSE for the Sobol sequence (s), Fixed design (f), Bin-based design (b), Maximin distance (md), Maximin weighted distance (wd), Maximum entropy (en), Integrated mean square error (ie), Maximum MSE (me), and Expected improvement (ei). Horizontal lines are the minimum and maximum errors from the Sobol sequences. Initial starting size of $n_0 = 16$ and final run size of $n = 64$.

Based on the plot in Figure 3 the two distance based criteria, the bin-based LHS and IMSE have the smaller RMSEs. However, when comparing maximum error the

two distance criteria and maximum entropy yield the smallest errors. The expected improvement criterion and maximum MSE are among the worst performers and are comparable to the Sobol design. The poor performance of maximum MSE is somewhat surprising; however, due to the optimization step in computing the criterion, one could get stuck in many local optima even though multiple restarts were used both in computing the criterion and in the design exchange algorithm. It is interesting to note that both of the distance criteria are performing as well or better than the other criteria. Indeed the computationally efficient version (weighted distance) of maximum MSE is on average performing much better than maximum MSE itself. The other striking feature to notice is the trade-off between maximum error and RMSE. In particular IMSE is the best performer in terms of RMSE but does not do near as well in terms of maximum error. In such cases, it is advisable to sacrifice some performance on RMSE to gain a much better performance in terms of maximum MSE. In this case the two distance criteria are performing extremely well.

Example 2: The values of $\boldsymbol{\theta}$ are chosen to be $(10, 3.6, 1.4)$, again a relatively hard problem but with a reasonable degree of sparsity, which helps improve the fit of the GP model as shown in Loeppky et al. (2008). Figure 4 shows results when using an initial starting design of $n_0 = 16$ runs and a final budget of $n = 64$ runs. The plot shows very similar results to the first example with the exception of the improved performance of maximin weighted distance compared to the maximin distance criterion. This is not particularly surprising considering that we expect weighted distance to perform better (Johnson et al., 1990) when the values of $\boldsymbol{\theta}$ are not all equal.

Example 3: The values of $\boldsymbol{\theta}$ are chosen to be $(11.25, 3.75, 0)$, which is a relatively hard problem with a reasonable degree of sparsity in that the three dimensional process is only active in two dimensions. This is one of the situations where the maximin weighted distance criterion could perform poorly due to the possibility of parameter estimates being zero. However, we will see that this is not the case especially when the criterion is implemented in a batch sequential fashion. In fact, one can often achieve a fairly uniform spread of points in the inactive dimension. In batch sequential optimization a random

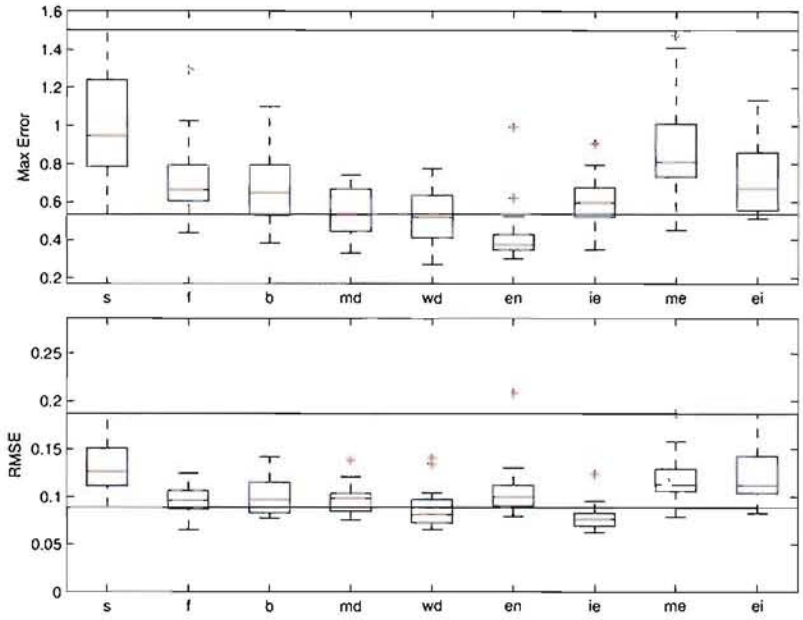

Figure 4: Generated data $\boldsymbol{\theta} = (10, 3.6, 1.4)$: Boxplots of maximum error and RMSE for the Sobol sequence (s), Fixed design (f), Bin-based design (b), Maximin distance (md), Maximin weighted distance (wd), Maximum entropy (en), Integrated mean square error (ie), Maximum MSE (me), and Expected improvement (ei). Horizontal lines are the minimum and maximum errors from the Sobol sequences. Initial starting size of $n_0 = 16$ and final run size of $n = 64$.

$m \times d$ matrix X_b is generated and updated using the Fedorov exchange. When $\theta_j = 0$ there is no gain to be made by updating the jth column vector of this matrix so that the spread of points tends to be fairly uniform.

The plot in Figure 5 shows the errors for each of the criteria starting from an initial design with $n_0 = 16$ points and final run size of $n = 64$. The plots show that the maximin weighted distance criterion and maximum entropy are outperforming all other methods in terms of both RMSE and maximum error, with the entropy criterion being slightly better than weighted distance.

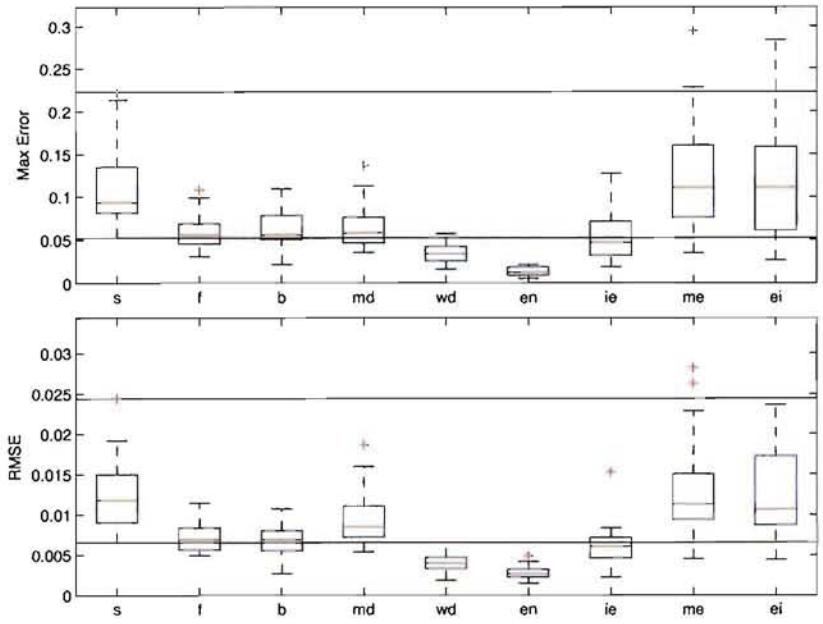

Figure 5: Generated data $\boldsymbol{\theta} = (11.25, 3.75, 0)$: Boxplots of maximum error and RMSE for the Sobol sequence (s), Fixed design (f), Bin-based design (b), Maximin distance (md), Maximin weighted distance (wd), Maximum entropy (en), Integrated mean square error (ie), Maximum MSE (me), and Expected improvement (ei). Horizontal lines are the minimum and maximum errors from the Sobol sequences. Initial starting size of $n_0 = 16$ and final run size of $n = 64$.

4.3 G-Protein Example

Yi et al. (2005) studied a computer model of ligand activation of G-protein in yeast. The computer code solves a system of ordinary differential equations (ODEs) with up to nine parameters that can vary, although we only vary four of these parameters in what follows. The response y is the normalized concentration of a relevant part of the G-protein complex after 30 seconds. The GP model is used to construct an approximation as a function of the log transformed input variables $\mathbf{x}$. The ODE solver is reasonably quick to run and enables us to evaluate various sequential strategies relatively quickly. In this situation the final budget was set at $n = 64$ runs and we considered the fixed design with

64 runs or starting with $n_0 = 16$ runs and adding points in $[0,1]^4$. The initial design is an OA-based Latin hypercube and is used as the starting design for each of the criteria. The underlying OA structure is a strength 2 OA for four 4-level factors constucted in similar fashion as the initial design in the 3-d examples using regular fractions in the eight factor 2-level design space and using a contractive replacement. As before, all of the methods except for the expected improvement criterion are implemented in a batch sequential fashion by selecting the best set of 8 runs in $[0,1]^4$. In addition we use Sobol sequences as the naive sequential strategy. In the bin-based LHS, the sequence of bins of 8 runs are translates associated with the initial bin set.

In order to induce some variability we take 24 starting designs for either n or n_0. These designs were constructed by taking all possible 4! permutations of the columns of the initial design. Due to the computational burden and the relatively poor performance of the maximum MSE criterion in the previous studies this criterion has been removed.

From the plot in Figure 6 we see that the maximin weighted distance criterion is performing well compared to the other criteria. Fixed designs and bin-based LHS are performing well and the Sobol sequence is the worst performer both on maximum error and RMSE.

4.4 PTW Code

We examine sequential augmentation with the PTW model (Preston et al., 2003) that describes plastic stress-strain relationships for metals. This implementation involves an 11-dimensional parameter space, with three design inputs (temperature, strain rate, strain) and eight tuning parameters.

We adopt a final run size of $n = 128$ and augment from an initial run size of $n_0 = 32$. The plot in Figure 7 shows the RMSE and maximum error for each of the design criteria considered.

The plot in Figure 7 shows that the maximin weighted distance criterion is performing reasonably well but not as well as a fixed design or a bin-based LHS in terms of RMSE.

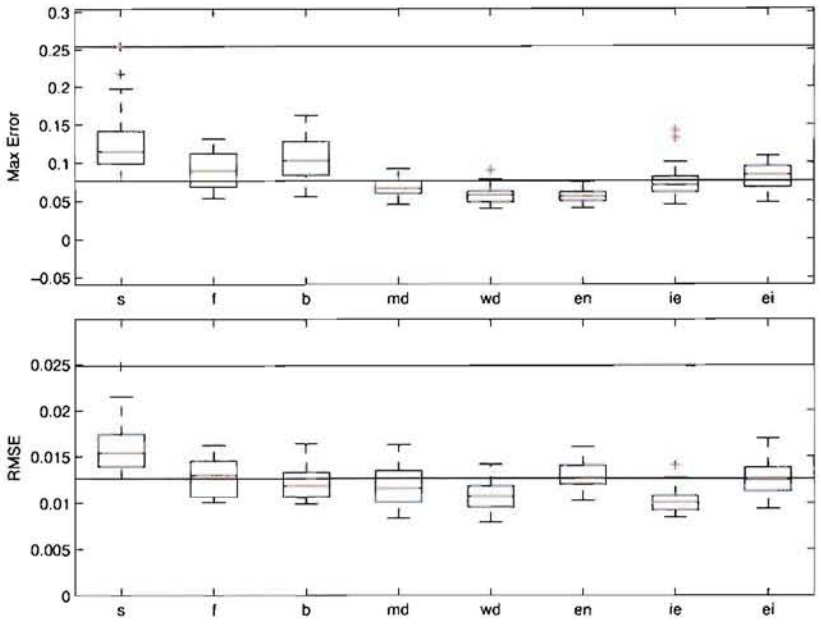

Figure 6: G-Protein Example: Boxplots of maximum error and RMSE for the Sobol sequence (s), Fixed design (f), Bin-based design (b), Maximin distance (md), Maximin weighted distance (wd), Maximum entropy (en), Integrated mean square error (ie), and Expected improvement (ei). Horizontal lines are the minimum and maximum errors from the Sobol sequences. Initial starting size of $n_0 = 16$ and final run size of $n = 64$.

Surprised by the poor performance of the maximin distance design in terms of RMSE, we investigated where the design points were being added and discovered that the majority of design points were being placed on the boundary of the space, a well known feature of the maximin distance criterion. To some extent the poor performance of the other criteria can also be explained by this phenomenon. The small initial run size may also play into the poor performance of the methods in that selection of the early batches of points may have been negatively influenced by a poor initial fit. In order to investigate this more throughly we also build the model starting from an initial design with $n_0 = 64$ runs, results are omitted. This did improve the overall performance of the criteria; however, it did not alleviate the problem of pushing points to the boundary. One promising

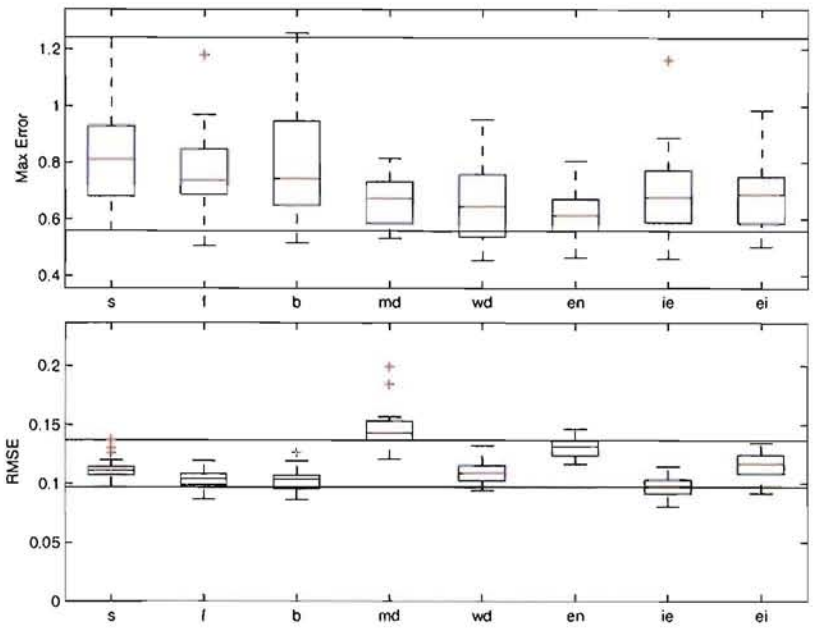

Figure 7: PTW Example: Boxplots of maximum error and RMSE for the Sobol sequence (s), Fixed design (f), Bin-based design (b), Maximin distance (md), Maximin weighted distance (wd), Maximum entropy (en), Integrated mean square error (ie), and Expected improvement (ei). Horizontal lines are the minimum and maximum errors from the Sobol sequences. Initial starting size of $n_0 = 32$ and final run size of $n = 128$.

direction we are currently investigating is design augmentation that combines bin-based LHS strategies with the design criteria of Section 3.

One final consideration for criteria selection is the computational cost of implementation. Table 2 shows the average time in seconds to add a batch of 8 runs to an existing design for the problems examined in the simulation study. In the case of $d = 3$ the average is computed over the 25 realizations for all three of the examples. It is easily seen that both the distance based criteria and maximum entropy are the most efficient in terms of computational cost. When coupling this with the overall performance in terms of RMSE and maximum error, maximin weighted distance or maximum entropy designs appear to the best choice for augmenting an initial design in order to achieve a good

	Distance	Weighted Distance	Entropy	IMSE	Maximum MSE
$d = 3$	4.12	3.51	4.42	15.01	537.31
$d = 4$	4.66	5.75	6.79	10.61	-
$d = 11$	23.33	16.07	30.27	52.64	-

Table 2: Average time in seconds to add a batch of 8 points to an existing design.

overall prediction of the model.

5 Discussion

In this paper we have discussed and implemented various strategies for the sequential adaptation of an initial design for fitting a GP model. In addition to the standard criteria used in the literature we have studied adding batches of new points based on two distance criteria (Johnson et al., 1990). Both of these criteria have performed well in smaller dimensional problems. In addition we introduce bin-based LHS designs which have shown great promise in terms of RMSE and performed well in higher dimensional examples. In most of the examples we included a fixed maximin LHS design, which performed as well as most of the sequential strategies in terms of RMSE, although the sequential strategies tend to provide better reductions in terms of maximum error. This highlights the commonly understood importance of filling the space, and suggests that major gains may be found in isolating the important effects and using a sequential strategy that fully explores the dimensions related to these factors while maintaining an adequate marginal distribution of points in the non-active factors. This topic is currently under investigation.

Although not studied in the article, an important consideration is in the number of runs to be used at the initial design stage. This is particularly true in higher dimensional examples where most of the criterion-based sequential methods did not perform well or provided very little gain over a fixed OA-based LHS design. In general, the best strategy

is to run an initial design that is sufficiently large to provide an acceptable estimate of the response surface, using any available prior knowledge regarding the behavior of computer model output in establishing an initial run size. If the experimental budget is fixed, at least a quarter of that budget should be expended on the initial design, with a greater proportion likely to be necessary in the presence of complex model behavior often associated with higher dimensional input spaces.

ACKNOWLEDGEMENTS

The research of Loeppky was supported by a grant from the Natural Sciences and Engineering Research Council of Canada. The research of Williams and Moore was supported by Cetin Unal of Los Alamos National Laboratory, through the Nuclear Energy Advanced Modeling and Simulation Campaign of the U.S. Department of Energy's Advanced Fuel Cycle Initiative. The authors also acknowledge the support and encouragement of C. C. Essix.

References

Aslett, R., Buck, R. J., Duvall, S. G., Sacks, J., and Welch, W. J. (1998), "Circuit Optimization via Sequential Computer Experiments: Design of an Ouput Buffer," *Applied Statistics*, 47, 31–48.

Bernardo, M. C., Buck, R., Liu, L., Nazaret, W. A., Sacks, J., and Welch, W. J. (1992), "Integrated Circuit Design Optimization Using a Sequential Strategy," *IEEE Transaction on Computer Aided Design*, 11, 361–372.

Blackwell, D. (1951), "Comparison of Experiment," in *Proceedings of the 2nd Berkeley Symposium*, Berkeley, Ca., University of California Press, pp. 93–102.

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*, 86, 953–963.

Fedorov, V. V. (1972), *Theory of Optimal Design*, New York: Academic.

Hedayat, A. S., Sloane, N. J., and Stufken, J. . (1999), *Orthogonal Arrays: Theory and Applications*, New York: Springer.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High-dimensional Output," *Journal of the American Statistical Association*, 103, 570–583.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004), "Combining Field Data and Computer Simulation for Calibration and Prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990), "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26, 131–148.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998), "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13, 455–492.

Lam, C. Q. and Notz, W. I. (2008), "Sequential Adaptive Designs in Computer Experiments for Response Surface Model Fit," Tech. rep., The Ohio State University.

Lindley, D. V. . (1956), "On a Measure of Information Provided by an Experiment," *Annals of Mathematical Statistics*, 27, 986–1005.

Loeppky, J. L., Sacks, J., and Welch, W. J. (2008), "Choosing the Sample Size of a Computer Experiment: A Practical Guide," Tech. rep., National Institute of Statistical Sciences.

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 Ouput from a Computer Code," *Technometrics*, 21, 239–245.

Mitchell, T. J. and Morris, M. D. (1992), "Bayesian Design and Analysis of Computer Experiments: Two Examples," *Statistica Sinica*, 2, 359–379.

Morris, M. D. and Mitchell, T. J. (1995), "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, 43, 381–402.

O'Hagan, A. (1992), "Some Bayesian Numerical Analysis," in *Bayesian Statistics 4*, eds. Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., Oxford University Press, pp. 345–363.

Owen, A. B. (1994), "Controlling Correlations in Latin Hypercube Samples," *Journal of the American Statistical Association*, 89, 1517–1522.

Preston, D. L., Tonks, D. L., and Wallace, D. C. (2003), "Model of Plastic Deformation for Extreme Loading Conditions," *Journal of Applied Physics*, 93, 211–220.

Ranjan, P., Bingham, D., and Michailidis, G. (2008), "Sequential Experiment Design for Contour Estimation from Complex Computer Codes," *Technometrics*, 50, 527–541.

Sacks, J. and Schiller, S. (1988), "Spatial Designs," in *Statistical Decision Theory and related Topics IV*, eds. Gupta, S. S. and Berger, J. O., Springer-Verlag, vol. 2, pp. 385–399.

Sacks, J., Schiller, S. B., and Welch, W. J. (1989a), "Designs for Computer Experiments," *Technometrics*, 31, 41–47.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989b), "Designs and Analysis of Computer Experiments (with Discussion)," *Statistical Science*, 4, 409–435.

Shewry, M. C. and Wynn, H. P. (1987), "Maximum Entropy Sampling," *Journal of Applied Statistics*, 14, 165–170.

Sobol, I. (1979), "On the Systematic Search in a Hypercube," *SIAM Journal of Numerical Analysis*, 16, 790–793.

Tang, B. (1993), "Orthogonal Array-based Latin Hypercubes," *Journal of the American Statistical Association*, 88, 1392–1397.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25.

Williams, B. J., Loeppky, J. L., Moore, L. M., and Macklem, M. S. (2009), "Batch Sequential Design to Achieve Predictive Maturity with Calibrated Computer Models," Tech. rep., Los Alamos National Laboratory.

Yi, T.-M., Fazel, M., Liu, X., Otitoju, T., Papachristodoulou, A., Prajna, S., and Doyle., J. (2005), "Application of Robust Model Validation Using SOSTOOLS to the Study of G-Protein Signaling in Yeast," in *Proceedings of Foundations of Systems Biology and Engineering*.