Generating Exact *D*-Optimal Designs for Polynomial Models

Jacob E. Boon
Operations Assessments Group
National Security Technology Department
The Johns Hopkins University Applied Physics Laboratory
Laurel, MD 20723
Email: jacob.boon@jhuapl.edu

Keywords: Optimal experimental design, mathematical optimization, general linear regression

Abstract

This paper compares several optimization algorithms that can be used to generate exact *D*-optimal designs (i.e., designs for a specified number of runs) for any polynomial model. The merits and limitations of each algorithm are demonstrated on several low-order polynomial models, with numerical results verified against analytical results. The efficiencies – with respect to estimating model parameters – of the *D*-optimal designs are also compared to the efficiencies of one commonly used class of experimental designs: fractional factorial designs. In the examples discussed, *D*-optimal designs are significantly more efficient than fractional factorial designs when the number of runs is close to the number of parameters in the model.

1. INTRODUCTION

In simulations and experiments where little is known about the underlying function relating the independent variables to the dependent variable, a polynomial model is often used to approximate the behavior of the computer simulation or real-world system. One common example of this approach is variable screening tests, where a multivariate linear model is used to identify the handful of independent variables – out of a potentially huge set of variables – that have a systematic effect on the dependent variable. Another application of polynomial models is response surface modeling, where a polynomial surface is used as an approximation of the true function in order to reduce the computational load of function optimization. In other contexts, there may be reason to believe that a particular polynomial model represents the true relationship between the inputs and output. The techniques discussed in this paper apply to all cases where a polynomial relationship between independent and dependent variables is assumed. Although the researcher may be interested in characterizing the bias error of the polynomial model (see [Goel et. al. 2006], for example), in many cases the goal of the experiment is simply to minimize the post-experiment uncertainty in the model parameters. In such cases, it is prudent to design the experiment based on the *D*-optimal criterion.

Polynomial models are of the form

$$z_k = \boldsymbol{h}_k^T \boldsymbol{\theta} + v_k \quad k = 1, \ldots, n,$$

where n is the number of simulation runs; v_k represents constant variance, mean-zero, uncorrelated process noise; $\boldsymbol{h}_k = \boldsymbol{g}(\boldsymbol{x}_k)$ is the p-dimensional design vector (which is a function of the vector input $\boldsymbol{x}$ for the k^{th} simulation run); $\boldsymbol{\theta}$ is a p-dimensional vector of regression coefficients; and z_k is the k^{th} simulation output. The form of the design vector $\boldsymbol{h}_k$ is determined by the specific regression model being used. For example, if the model includes an intercept, linear term, and quadratic term for a single factor (independent variable), then $\boldsymbol{x}_k$ will be a scalar and $\boldsymbol{h}_k = [1 \; x_k \; x_k^2]^T$. In order to be a polynomial model, $\boldsymbol{h}_k$ must consist only of multiplicative combinations of the factors to various powers (e.g., x_1^3, x_3, $x_2x_5^2$, etc.). The n input vectors are chosen from $N \leq n$ design points (X_1, X_2,…, X_N) in the space of possible input combinations. Although we will consider only hypercube design spaces, some of the techniques considered in this paper can easily be extended to more general design spaces that one might encounter in practice. The choice of input vectors at various design points is summarized by the concept of a design, denoted

$$\xi \equiv \begin{Bmatrix} X_1 & X_2 & \cdots & X_N \\ w_1 & w_2 & \cdots & w_N \end{Bmatrix},$$

where w_i represents the proportion of total runs executed at design point X_i. Being a set of proportions, we constrain the w_i to satisfy

$$\sum_{i=1}^{N} w_i = 1.$$

For finite-sample designs, note that $w_i n$ must be a natural number.

In this context, one of the most common goals of experimental design is to identify a design ξ that minimizes the post-experiment uncertainty in the p parameter estimates. More formally, the goal is to choose ξ such that volume of the p-dimensional confidence ellipsoid for $\boldsymbol{\theta}$ is minimized. This can be accomplished by choosing ξ such

 ISBN 1-56555-314-4

that $\det[\boldsymbol{H}^T\boldsymbol{H}]$ is maximized, where $\boldsymbol{H}$ is the $n \times p$ matrix with the i^{th} row equal to $\boldsymbol{h}_i$ (see, for example, [Spall 2003] pp. 471-472). A design that maximizes $\det[\boldsymbol{H}^T\boldsymbol{H}]$ for a fixed number of runs is referred to as an exact *D*-optimal design. Regardless of the specific form of the regression model (i.e., regardless of the specific form of the $\boldsymbol{h}_k$), the matrix $\boldsymbol{H}$ has $d = nf$ degrees of freedom, where f is the number of factors. Thus, the problem of choosing the best design can be cast as a minimization problem, where the dimension of the search space is d and with loss function $L(\boldsymbol{\mu}) = -\det[\boldsymbol{H}^T\boldsymbol{H}]$ ($\boldsymbol{\mu}$ is the d-dimensional vector of input levels for each factor in each run).

Even for relatively small n and f, the dimension of the search space d quickly becomes quite large. For example, if $n = 10$ and $f = 2$, then $d = 20$. This makes identifying a global minimum of $L(\boldsymbol{\mu})$ for large n and large f difficult. The challenge of obtaining an exact *D*-optimal design for general n, f, and polynomial regression model is one of the reasons researchers rely so heavily on classical methods of experimental design (e.g.,. fractional factorial designs). In the following sections, we examine a few specific examples that test the limits of several optimization algorithms in identifying *D*-optimal experimental designs.

2. OPTIMIZATION ALGORITHMS

The application of numerical techniques to the problem of identifying exact *D*-optimal designs is not new. Chapter 15 of [Atkinson and Donev 1992] describes several algorithms that have been applied in this context. One such class of algorithms, called exchange algorithms, involves three steps: (1) the generation of a list of M candidate points in the design space, (2) the generation of an initial design typically based on the M candidate points, and (3) iterative updates of the initial design through changes in the design weights w_i ($i = 1,\ldots,M$) with the goal of improving the design with respect to the *D*-optimal criterion. Matlab's statistics toolbox contains two implementations of such exchange algorithms: *rowexch* and *cordexch*. Both functions generate the same default list of candidate points: the set of 3^f factorial points at levels −1, 0, and 1 for quadratic models, and the set of 2^f factorial points at levels −1 and 1 for linear models. Polynomial models of order greater than order two are not handled by *rowexch* and *cordexch*. The default initial design for both functions is generated by selecting each of the d inputs from a uniform distribution over [−1,1]. This represents a departure from most exchange algorithms, which typically choose an initial design based on the list of carefully chosen candidate points. The two Matlab functions differ, though, as to how the initial design is updated. In the case of *rowexch*, the algorithm evaluates the utility of replacing each point in the initial random design by one of the points in the set of M candidate design points. The algorithm sequentially replaces each of the design points by the candidate design point that leads to the largest gain in $\det[\boldsymbol{H}^T\boldsymbol{H}]$, if any gain is possible. The exchange portion of *cordexch* works similarly, although this algorithm works at the component level, treating each of the f inputs in each of the n runs separately.

In addition to these exchange algorithms, we will consider three well-known optimization algorithms: Blind Random Search (BRS), Localized Random Search (LRS), and Simultaneous Perturbation Stochastic Approximation (SPSA).[1] Knowing that the efficiency of BRS is poor even when the dimension of the search space is only moderately high (e.g., more than four or five in many applications), this algorithm serves primarily as a baseline upon which to improve. LRS search is well-suited to a problem of this type, where loss function evaluations are noise-free and the appropriate choice of random search vectors ensures that one will not get stuck at a local minimum. It is also natural to apply SPSA in this context, since the randomness in search direction and magnitude, like LRS, ensures that SPSA will not get stuck at a local minima. SPSA has the additional feature of using a noisy approximation of the loss function's gradient to guide the direction of its search. Note that although SPSA's stepwise approximation of the loss function is noisy, each individual loss function measurement is noise-free.

Before discussing results, a few words about the particular implementation of LRS and SPSA used here. LRS works by perturbing its estimate of the optimal $\boldsymbol{\theta}$ using a random search vector at each iteration and checking for improvement in the loss function. To allow LRS converge more rapidly in the latter stages of its search, the variance of the random search vector generated at each iteration is reduced from 0.025 to 0.01 after 70% of its computation budget is exhausted. Similarly, the parameters that control how fast SPSA's gain sequences decay are changed from the standard $\alpha = 0.602$ and $\gamma = 0.101$ to the asymptotically

[1] For details on these algorithms, see [Spall 2003] Chapter 2 (BRS and LRS) and Chapter 7 (SPSA).

Table 1. $\det[\boldsymbol{H}_{alg}{}^{T}\boldsymbol{H}_{alg}] / \det[\boldsymbol{H}_{optimal}{}^{T}\boldsymbol{H}_{optimal}]$ for various approaches. Results apply to a quadratic regression with two factors.

n	BRS	LRS	SPSA	SPSA best	*rowexch*	*cordexch*
6	0.190	0.999	0.996	1.000	0.956*	0.983
7	0.163	0.999	0.996	1.000	0.979*	0.994
8	0.172	0.997	0.991	1.000	0.976*	0.997
9	0.130	0.999	0.973	0.999	1**	1**

* Equivalent to a fractional factorial design; ** Equivalent to a full factorial design

optimal $\alpha = 1$ and $\gamma = 1/6$ after 70% of its computational budget is exhausted.[2] Since the loss function associated with SPSA's estimate is not, in general, monotonically increasing from iteration to iteration, it is possible that SPSA may happen across the optimal design (or a design arbitrarily close to it) and then proceed to search among lesser designs in future iterations. For this reason, we will also consider the best design SPSA generates across all iterations. This estimate will be referred to as 'SPSA best,' and computing it requires making an extra loss function measurement at each iteration of SPSA. Apart from the few exceptions noted here, the implementations of BRS, LRS, and SPSA used in this paper are standard.

3. QUADRATIC REGRESSION WITH TWO FACTORS

In this section, we consider the most general second-order polynomial, but in order to compare with known solutions, we limit ourselves to two factors. Thus, our regression model is

$$z = [1, x_1, x_1^2, x_2, x_2^2, x_1 x_2]\boldsymbol{\theta} + v,$$

where x_1 and x_2 are scalars such that $-1 \le x_1, x_2 \le 1$,[3] and $\boldsymbol{\theta}$ is a 6-dimensional vector of regression coefficients.

[Box and Draper 1971] solved the problem of generating an exact *D*-optimal design for this regression model when the number of simulation runs $n = 6, 7, \ldots, 18$. According to these authors, the *D*-optimal designs for each *n* were obtained via a computer hill-climbing search. Exact *D*-optimal designs for $n = 6, \ldots, 9$ are as follows:

$n = 6$: $(-1, -1)$, $(1, -1)$, $(-1, 1)$, $(-\delta, -\delta)$, $(1, 3\delta)$, $(3\delta, 1)$, where $\delta = (4-\sqrt{13})/3 = 0.1315$;
$n = 7$: $(\pm 1, \pm 1)$, $(-0.092, 0.092)$, $(1, -0.067)$, $(0.067, -1)$;
$n = 8$: $(\pm 1, \pm 1)$, $(1, 0)$, $(0.082, -1)$, $(0.082, -1)$, $(-0.215, 0)$;
$n = 9$: the 3^2 factorial at levels -1, 0, and 1.

For their own independent searches, BRS, LRS, and SPSA were each given a sizeable budget of 10^6 (1 million) loss function evaluations. Note that, since the computation of 'SPSA best' requires an extra loss function measurement at each iteration, 'SPSA best' is really the result of 1.5 million loss function evaluations. Since the number of loss function measurements made during each call to *rowexch* and *cordexch* varies, it is impossible to assign them the same fixed computational budget. To cope with this, we store the best design across 10,000 replications of *rowexch* and *cordexch*. In this way, the runtime of each of the five algorithms is roughly the same. The results of these searches are summarized in Table 1.

Although *rowexch* is able to incorporate design points in its initial random design (generated in exactly the same way that BRS generates its candidate designs), it never did so in this experiment. Thus, out of 10,000 initial random designs, none of the design points were better suited to the problem of estimating the optimal $\boldsymbol{\theta}$ than the points in the 3^2 factorial space. As a result, *rowecxh* simply generated the best fractional factorial design for each *n*. On the other hand, the component-wise exchange in *cordexch* did find design points better suited for estimating $\boldsymbol{\theta}$. In this way, *cordexch* generated designs more efficient than the fractional factorial designs.

As expected, BRS is unable to cope with the high dimensionality of the search space (in this example, the dimension is between 12 and 18). On the other hand, there is nothing stopping BRS from stumbling upon an excellent design – it just may take billions (or more!) iterations. We also observe that LRS performs very well, considering the lack of directionality in its search. In fact, LRS outperforms

[2] See [Spall 2003] p. 190 for a discussion of reasonable choices for α and γ, including the recommendation to convert α and γ to their asymptotically optimal values at some point in the search.

[3] The *D*-optimal criterion has the desirable property of transform invariance. That is, a *D*-optimal design for each factor ranging between -1 and 1 is also a *D*-optimal design after the range of each factor undergoes a linear transformation (with design points scaled in the same way, of course).

 ISBN 1-56555-314-4

Table 2. $\det[\boldsymbol{H}_{\text{alg}}{}^{\text{T}}\boldsymbol{H}_{\text{alg}}] / \det[\boldsymbol{H}_{\text{optimal}}{}^{\text{T}}\boldsymbol{H}_{\text{optimal}}]$ for various approaches. Results apply to a quadratic regression with 3 factors.

n	BRS	LRS	SPSA	SPSA best	*rowexch*	*cordexch*
10	0.000	0.960	0.994	0.996	0.716	0.716

SPSA, unless you consider the best SPSA design across all iterations. Although SPSA and LRS provided more efficient designs than Matlab's exchange algorithms in some cases, the difference is not extraordinary. For this particular example, using the fractional factorial designs – which are easy to generate – only slightly degrades the potential efficiency of the design. Still, one might imagine cases where each experiment is so costly (or time-consuming) that refining a fractional factorial design is worthwhile. We do note, however, that the gap in design efficiency between fractional factorial design and the exact D-optimal design is largest when the number of runs is equal to the number of parameters in the model.

4. QUADRATIC REGRESSION WITH THREE FACTORS

Extending the results of the previous section, we now consider the problem of generating an exact *D*-optimal design for the quadratic regression model with three factors

$$z = [1, x_1, x_1^2, x_2, x_2^2, x_3, x_3^2, x_1x_2, x_1x_3, x_2x_3]\boldsymbol{\theta} + v$$

and n = 10 simulation runs. Note that since the model contains ten parameters, the number of simulation runs must be at least ten. In this setting, the dimension of the design space is nf = 30. [Box and Draper 1971] also solved this problem, stating that the optimal inputs for ten runs are

(–1,–1,–1), (1,–1,–1), (–1,1,–1), (–1,–1,1), (–1,α,α), (α,–1,α), (α,α,–1), (–β,1,1), (1,–β,1), and (1,1,–β), where α = 0.1925 and β = 0.2912.

Applying the same algorithms as in the previous sections, again allowing 10^6 loss function evaluations for BRS, LRS, and SPSA, as well as 10,000 replications of *rowexch* and *cordexch*, we get the results summarized in Table 2. Again we observe the futility of BRS in a high-dimensional situation; even with 10^6 iterations, the BRS design fails to have even 0.1% of the efficiency of the optimal design. We also observe that LRS and SPSA are able to generate designs very close to the exact *D*-optimal design, with the best SPSA design across all iterations coming within 0.4% of the exact *D*-optimal design.

Since the regression model used in this example (and the last) is quadratic in the factors, fractional factorial designs consist of the factors set at one of three levels – either –1, 0, or 1 – in each run. Thus, with 30 input levels to choose, there are $3^{30} \approx 2 \times 10^{14}$ possible fractional factorial designs. Although *rowech* and *cordexch* are not constrained to generate fractional factorial designs, they both returned factional factorial designs in this case. With the gap in design efficiency between good fractional factorial designs (generated by *rowech* and *cordexch*) and the exact *D*-optimal design at around 28%, this is a case where allowing input levels to take on more than two or three values (*a la* many classical experimental designs) offers a significant gain in performance.

5. PURE CUBIC REGRESSION WITH ONE FACTOR

In this final example, we show that numerical methods can also be helpful in determining *asymptotic D*-optimal designs. An asymptotically *D*-optimal design, as opposed to an exact *D*-optimal design, assigns weights w_i to the N design points assuming an infinite number of runs. We investigate the optimum experimental design for the regression model

$$z = [1, x, x^2, x^3]\boldsymbol{\theta} + v,$$

where $-1 \le x \le 1$ is a scalar input. In this case, $\boldsymbol{\theta}$ contains four parameters, so the number of runs n must be at least four. We proceed by finding the optimal finite-sample design with n = 4 and showing via the Kiefer-Wolfowitz equivalence theorem that this design is in fact the *D*-optimal asymptotic design.

To test design ξ for asymptotic efficiency using the Kiefer-Wolfowitz equivalence theorem (see [Spall 2003] p. 478 for a statement of the theorem), we must compute the variance function

$$V(x, \xi) = \boldsymbol{g}(x)^{\text{T}}\, \boldsymbol{M}(\xi)^{-1}\, \boldsymbol{g}(x),$$

where $\boldsymbol{g}(x)$ maps inputs to the space of design vectors and $\boldsymbol{M}(\xi)$ is the precision matrix

$$\boldsymbol{M}(\xi) = \sum_{i=1}^{N} w_i\, \boldsymbol{g}(X_i)\boldsymbol{g}(X_i)^{\text{T}}.$$

The variance function $V(x,\xi)$ is proportional to the variance of predictions $\hat{z}(x)$, and if $V(x,\xi)$ is less than p for all x given design ξ, the Kiefer-Wolfowitz equivalence theorem states that ξ is an asymptotically *D*-optimal design.

One might approach the problem of identifying the *D*-optimal asymptotic design by first identifying a candidate

set of support points and then searching for the optimal weighting across these points. This approach is analogous to that of factorial designs, where the factors are constrained to take on either –1 or 1 for linear regression models and either –1, 0, or 1 for quadratic regression models. In the current example, if the candidate set of support points is specified to be {–1, –0.5, 0, 0.5, 1}, the best design across the $5^4 = 625$ possible designs is

$$\xi^{(1)} = \begin{Bmatrix} -1 & -0.5 & 0.5 & 1 \\ ¼ & ¼ & ¼ & ¼ \end{Bmatrix}$$

For this design, the value of $\det[\boldsymbol{H}^T\boldsymbol{H}]$, where $\boldsymbol{H}$ is one of the 4! = 24 possible permutations of the four design points, is 1.2656.

Computing the variance function $V(x,\xi^{(1)})$ and plotting $V(x,\xi^{(1)})$ versus x ($-1 \le x \le 1$) gives Figure 1.

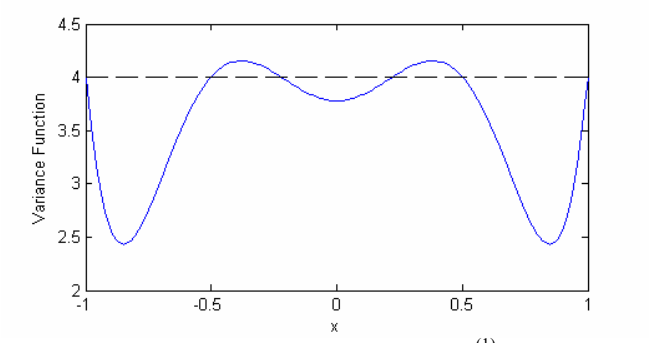

Figure 1. Plot of the variance function $V(x,\xi^{(1)})$.

Since $V(x,\xi^{(1)})$ is greater than $p = 4$ for some x, the Kiefer-Wolfowitz equivalence theorem ensures that $\xi^{(1)}$ is not the *D*-optimal asymptotic design. On the other hand, the maximum of the variance function is not much greater than p (it is approximately 4.152), which suggests that $\xi^{(1)}$ is "close" to the *D*-optimal asymptotic design.

Another approach is to perform a numerical search, such as has been described in this paper, over the continuous domain $[-1,1]^4$ in $\mathbf{R}^4$. The following design was generated with 250,000 iterations of SPSA:

$$\xi^{(2)} = \begin{Bmatrix} -1 & -0.4472 & 0.4420 & 1 \\ ¼ & ¼ & ¼ & ¼ \end{Bmatrix}.$$

For this design, the value of $\det[\boldsymbol{H}^T\boldsymbol{H}]$ is 1.3105. Furthermore, computing the variance function $V(x,\xi^{(2)})$ and plotting $V(x,\xi^{(2)})$ versus x where $-1 \le x \le 1$ gives Figure 2.

In addition to $\det[\boldsymbol{H}^T\boldsymbol{H}]$ being larger for $\xi^{(2)}$ than for $\xi^{(1)}$, $\xi^{(2)}$ is very close to satisfying the condition that $V(x,\xi^{(2)}) \le p$ (the maximum value of $V(x,\xi^{(2)})$ is 4.001 at x = 0.45). Thus, in terms of efficiency in the choice of design points, $\xi^{(2)}$ is very close to a *D*-optimal asymptotic design.

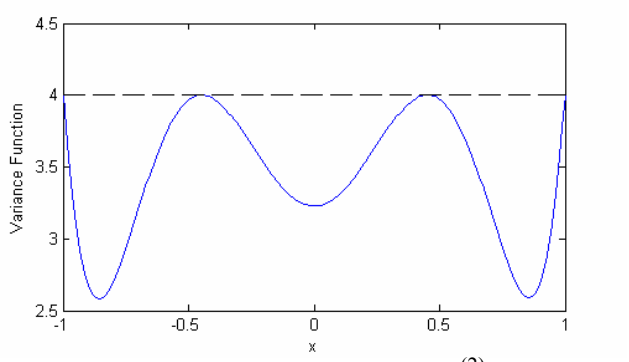

Figure 2. Plot of the variance function $V(x,\xi^{(2)})$.

A third approach is to attack the problem analytically. If we assume that the set of optimal design points is of the general form $\{a_1, a_2, a_3, a_4\}$, where $-1 \le a_i \le 1$ (i = 1,…,4), then $\det[\boldsymbol{H}^T\boldsymbol{H}]$ does not have enough structure to determine the optimal values for the a_i. On the other hand, if we assume that the set of optimal design points is of the form $\{-1, -a, a, 1\}$, as suggested by the numerical search, $\det[\boldsymbol{H}^T\boldsymbol{H}]$ has the form

$$16a^{10} - 64a^8 + 96a^6 - 64a^4 + 16a^2 .$$

Setting the derivative of this equal to zero, we get

$$5a^9 - 16a^7 + 18a^5 - 8a^3 + a = 0 .$$

Factoring this result, we get

$$a(5a^2 - 1)(a-1)^3(a+1)^3 = 0 .$$

The above polynomial in a has roots at –1, 1, 0, $1/\sqrt{5}$, and $-1/\sqrt{5}$. Inspection of the second derivative or a plot of $\det[\boldsymbol{H}^T\boldsymbol{H}]$ reveals that only $1/\sqrt{5}$ and $-1/\sqrt{5}$ provide local maxima. Note that $1/\sqrt{5} \approx 0.44721$, which makes the following design very similar to the one found via the numerical search:

$$\xi^{(3)} = \begin{Bmatrix} -1 & -1/\sqrt{5} & 1/\sqrt{5} & 1 \\ ¼ & ¼ & ¼ & ¼ \end{Bmatrix}.$$

A plot of $V(x,\xi^{(3)})$ reveals that the maximum value of the variance function over the interval $-1 \le x \le 1$ is $p = 4$. Thus, by the Kiefer-Wolfowitz equivalence theorem, design $\xi^{(3)}$ is a *D*-optimal asymptotic design. In this example, the numerical approach was instrumental in simplifying the analytical optimization of $\det[\boldsymbol{H}^T\boldsymbol{H}]$.

6. CONCLUSIONS

We have explored several techniques that can be used to numerically search for exact *D*-optimal designs. For the regression models considered, we observed that BRS was

unable to cope with the high-dimensionality of the search space. LRS and SPSA, on the other hand, did converge to the optimal design after many iterations. But because the magnitude of $\det[\boldsymbol{H}_{\text{optimal}}{}^{T}\boldsymbol{H}_{\text{optimal}}]$ for the optimal design matrix $\boldsymbol{H}_{\text{optimal}}$ increases as the number of runs n and factors f increases, SPSA needed to be retuned for each problem. LRS, on the other hand, did not require any retuning; that is, the same distribution of random search vectors was used in all cases. Another desirable property of LRS and SPSA is that they can be easily extended to more general design spaces (i.e., space more complex than a hypercube). Matlab's exchange algorithms can also be extended in this way, although the user must generate custom candidate lists and initial designs that adhere to the design space constraints.

This paper did not explore the role of orthogonality in the selection of experimental designs. If a design is orthogonal, then each component of $\boldsymbol{\theta}$ may be treated independently from the others in post-experiment hypothesis testing. In many cases, the designs generated by Matlab's exchange algorithms were, for the most part, orthogonal; the nearly *D*-optimal designs generated by LRS and SPSA, on the other hand, would typically lead to undesirable correlations between components of $\boldsymbol{\theta}$. The tradeoff – if there is one to be made – between orthogonality and *D*-optimality must be made by the researcher. If orthogonality or near-orthogonality is required, the necessary constraints could be imposed on SPSA and LRS to generate such designs.

This work might be extended by investigating the performance of these techniques when the number of factors is considerably higher. Although analytical solutions have not been derived in many cases, numerical techniques could at least be compared with fractional factorial designs. Another extension of this work might be a theoretical comparison, possibly within the framework suggested by [Spall et. al. 2006], of algorithm efficiencies for finding *D*-optimal designs in certain classes of linear regression models. Finally, one might explore the efficiencies of these techniques when generating designs based on other design criteria, such as the model-robust and model-sensitive criteria discussed in [Goos et. al. 2005].

References

Atkinson, A. C. and Donev, A. N. (1992). *Optimum Experimental Designs*. Oxford University Press, Oxford.

Box, M. J. and Draper, N. R. (1971). "Factorial Designs, the $|X^T X|$ Criterion, and Some Related Matters." *Technometrics Vol. 13 Issue 4*. 731-742.

Goel, T., Haftka, R. T., Papila, M., Shyy, W. (2006). "Generalized Pointwise Bias Error Bounds for Response Surface Approximations." *International Journal for Numerical Methods in Engineering Vol. 65 Is. 12*. 2035-2059.

Goos, P., Kobilinsky A., O'Brien T. E., and Vandebroek, M. (2005). "Model-Robust and Model-Sensitive Designs." *Computational Statistics & Data Analysis Vol. 49*. 201-216.

Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Spall, J. C., Hill, S. D., and Stark, D. R. (2006). "Theoretical Framework for Comparing Several Stochastic Optimization Approaches," in *Probabilistic and Randomized Methods for Design under Uncertainty* (G. Calafiore and F. Dabbene, eds.). Springer-Verlag, London, Chapter 3 (pp. 99-117).

Biography

Since graduating from Cornell University in 2003 with a Bachelor's degree in Mathematics and Philosophy, Jacob Boon has been a member of the Associate Professional Staff at the Johns Hopkins University Applied Physics Laboratory. In 2006, Mr. Boon earned a Master's degree in Applied and Computational Mathematics from Johns Hopkins University.