Robust Experimental Design for Multivariate Generalized Linear Models

Hovav A. DROR and David M. STEINBERG

Department of Statistics and Operations Research
Raymond and Beverly Sackler Faculty of Exact Sciences
Tel Aviv University
Ramat Aviv 69978
Israel
(*hovavdror@gmail.com*; *dms@post.tau.ac.il*)

A simple heuristic is proposed for constructing robust experimental designs for multivariate generalized linear models. The method is based on clustering a set of local optimal designs. A method for finding local D-optimal designs using available resources is also introduced. Clustering, with its simplicity and minimal computation needs, is demonstrated to outperform more complex and sophisticated methods.

KEY WORDS: Binary response; Clustering; D-optimal; Design of experiments; Logit; Poisson.

1. INTRODUCTION

Efficient experimental designs for generalized linear models (GLMs) depend on the unknown coefficients; therefore—unlike experimental designs for linear models—two experiments having the same model but different coefficient values will typically require different designs. For any given set of values for the model parameters, there is an experimental design that is locally optimal. However, because there is uncertainty about the values, one should look for an experimental design that performs well all over the uncertainty space, giving higher priority to regions of higher likelihood within that space.

Previous work on experimental designs for GLMs is focused mainly on locally optimal designs for a simple linear effect and one design variable (see, e.g., Abdelbasit and Plackett 1983; Ford, Torsney, and Wu 1992; Mathew and Sinha 2001). Most extensions (e.g., Sitter and Torsney 1995a, b) are limited to two factors or to first-order models that do not contain interactions. The limited amount of research is due in part to the fact that finding locally optimal designs for GLMs, and even more so for high-order multivariate models, is far from a trivial task. Section 2 describes a fast, simple method for finding local D-optimal designs for these complex cases.

The ability to find locally optimal designs still leaves the serious problem of how to take into account uncertainty with respect to model coefficients. Different attitudes toward design robustness for univariate GLMs have been expressed by Abdelbasit and Plackett (1983), Sitter (1992), Hedayat, Yan, and Pezzuto (1997), and Chaloner and Larntz (1989). Of these approaches, the latter should be emphasized for suggesting a Bayesian experimental design. Literature on multivariate robust designs for GLMs is scarce and includes an unpublished manuscript by Chipman and Welch (available at *http://ace.acadiau.ca/math/chipmanh/papers/chipman-d-opt.ps*) that suggests a minimax approach and an article by Robinson and Khuri (2003) that evokes the idea of using so-called "quantile dispersion" graphs. Khuri, Mukherjee, Sinha, and Ghosh (2004, p. 42) surveyed design issues for GLMs and noted that "the research on designs for generalized linear models is still very much in developmental stage. Not much work has been accomplished either in terms of theory or in terms of computational methods to evaluate the optimal design when the dimension of the design space is high. The situation when one has several covariates. . . demand[s] extensive work to evaluate "optimal or at least efficient designs." Recently Woods, Lewis, Eccleston, and Russell (2006) delivered much of the sought-after results by proposing a method for finding multivariate compromise designs that allow for uncertainty in the link function, the linear predictor, or the model parameters.

In this article we suggest a simple heuristic capable of finding designs that are robust to most parameters an experimenter might consider, including (similar to Woods et al. 2006) uncertainty in the coefficient values, in the linear predictor equation, and in the link function. Compared with Bayesian designs, such as those of Chaloner and Larntz (1989), or the compromise designs of Woods et al. (2006), the suggested procedure requires considerably shorter computation time and is easier to implement, requiring only the ability to find locally optimal designs and a K-means clustering procedure (MacQueen 1967). The process allows rapid exploration of various designs, enabling the procedure to outperform the existing alternatives.

Given a set of local D-optimal designs, the core of the method proposed is to combine them into a set of location vectors and use K-means clustering to derive a robust design, as motivated by the following examples.

1.1 Example 1

Assume a logistic model with the linear predictor $\eta = \beta_0 + \beta_x x + \beta_y y + \beta_{xy} xy$ having uncertainty about β_0 modeled as a uniform distribution over the region $[0, 2]$ with $\beta_x = \beta_y = 2$, $\beta_{xy} = .2$. Figure 1 shows the local D-optimal designs for this model, for 25 different equally spaced values of β_0 from the feasible region.

Each local D-optimal design has four support points. It can be seen that different values of β_0 result in a small change of the location of these support points, and as a result there is a clear

TECHNOMETRICS, NOVEMBER 2006, VOL. 48, NO. 4
DOI 10.1198/004017006000000318

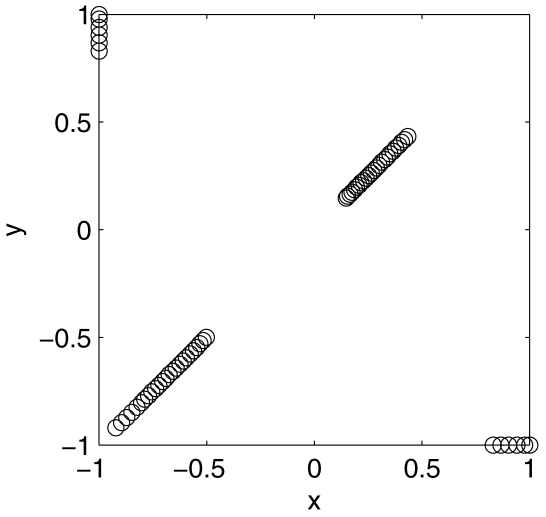

Figure 1. Proximity of 25 Local D-Optimal Designs for a Logistic Model With Intercept Value Uncertainty.

partition of the local designs' points to four clusters. Desiring an efficient experimental design without knowing any further information, it seems reasonable to place one point at the "middle" of each cluster.

1.2 Example 2

Woods et al. (2006) noted that the local D-optimal design for the centroid of the β's uncertainty space often may be an efficient compromise design. A preference for clustering can be justified only if it remains an efficient method even in conditions where using the best local D-optimal design fails to perform well.

Continuing Example 1 but assuming a larger uncertainty region for β_0 causes the four clusters to overlap. Figure 2 displays local D-optimal designs for 25 different values of β_0 from $[0, 15]$ with $\beta_x = \beta_y = 10$ and $\beta_{xy} = .2$. The filled points in the figure show the local D-optimal design for the centroid of the feasible region, $\beta_0 = 7.5$.

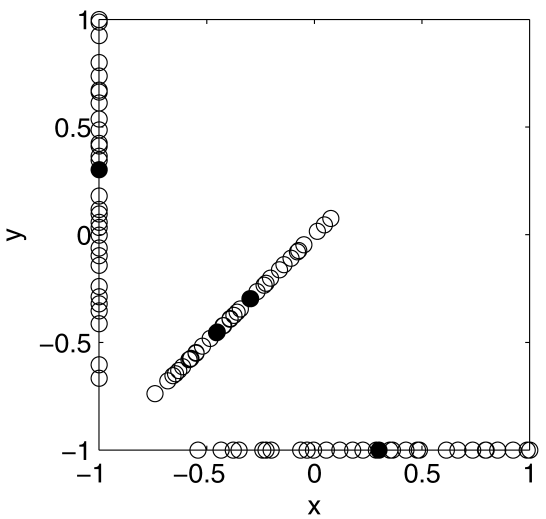

Figure 2. An Illustration of the Shortcoming of the Best Local D-Optimal Design.

It is seen that the local D-optimal design for the centroid of the beta space has two support points on the diagonal whose distance from each other is smaller than the range of diagonal point shifts for other possible values of β_0. Coverage of the design space through clustering has better potential for creating a robust design than the parameter space centroid or any other local D-optimal design.

2. FINDING LOCAL D–OPTIMAL DESIGNS

The procedure suggested in this article assumes the ability to easily construct local optimal designs. The assumption is far from being trivial, because common packages such as "gosset" (Hardin and Sloane 1993), the statistical toolbox in MATLAB (MathWorks Inc.), JMP, or the SAS Optex procedure were not designed to be used with GLMs.

Finding an exact local D-optimal design for GLMs requires finding a choice of n support points that will maximize the determinant of the information matrix. For linear models, the information matrix is simply $\mathbf{F}'\mathbf{F}$, where $\mathbf{F}$ is the regression matrix. For GLMs, the information matrix also depends on a weight matrix and can be represented as $\mathbf{F}'\mathbf{W}\mathbf{F}$ (see, e.g., Atkinson and Donev 1992). The weights are given by $\mathbf{W} = V^{-1}(\boldsymbol{\mu})(d\boldsymbol{\mu}/d\boldsymbol{\eta})^2$, where V is the variance function; $\boldsymbol{\mu}$ is a vector with row values, μ_i, being the expected response for the experimental configuration expressed by the row $\mathbf{F}_i$ of the regression matrix; $\boldsymbol{\eta} = \mathbf{F}\boldsymbol{\beta}$ is the linear predictor; $\boldsymbol{\beta}$ is the vector of p unknown coefficients; and the relation between μ_i and η_i is expressed through a given link function. For example, for a Poisson model with a log link, the diagonal elements of W are $w_{ii} = \mu_i = \exp(\mathbf{F}_i\boldsymbol{\beta})$, and for a binary response with the logit link, $w_{ii} = \mu_i(1-\mu_i) = \exp(\mathbf{F}_i\boldsymbol{\beta})/(1+\exp(\mathbf{F}_i\boldsymbol{\beta}))^2$.

Thus, given the values of $\boldsymbol{\beta}$, we can compute the values of the diagonal matrix $\mathbf{W}$ for any candidate set of design points. Thus, local D-optimal designs for generalized linear models can be found by setting $\tilde{\mathbf{F}} = \mathbf{F}\sqrt{W}$ and using a row-exchange algorithm, such as that of Federov (1972), to find an n point subset of $\mathbf{F}$ that maximizes the determinant of the information matrix $\tilde{\mathbf{F}}'\tilde{\mathbf{F}}$.

For multivariate problems, a good candidate set may be of enormous size, causing common computer algorithms to malfunction or preventing their implementation. To overcome this obstacle, sequential methods may be used. Begin with a rough grid chosen at random or from a low-discrepancy sequence. [A short description of low-discrepancy (also known as quasi-random) sequences is given in the App.] For this candidate set, calculate the regression matrix and find a D-optimal design. Use the result to create a new candidate set, with each support point of the D-optimal design found being the center of a new random or quasi-random sequence. To avoid large candidate sets, limit the size of the sequence around each point so that the number of candidate points from all sequences will be reasonably small; in the examples presented herein, we used 50 normally distributed points around each candidate. Create a rule for adjusting the search radius around the points; for instance, reduce the search radius according to the largest distance between points in the new design compared with the previous step, but no less than 30% of the last search radius used. Create a stopping rule in accordance with the accuracy desired.

To give an idea of the effectiveness of the procedure just described, it takes less than 1 second to produce a 16-point local D-optimal design accurate to 2 decimal places for the 5-variable Poisson model containing 2 interactions used in Section 6. Computer run times presented in this article were measured using a desktop PC with a 2.4-GHz Celeron processor.

An implementation of the algorithm and procedures for examples from the following sections are available at *http://www.math.tau.ac.il/~dms/GLM_Design*.

3. CLUSTERING VERSUS BAYESIAN DESIGNS

Chaloner and Larntz (1989) discussed construction of Bayesian-optimal designs for a one-variable (two parameters) logistic regression, where the probability of success for an observation at $x \in [-1, 1]$ is $p(x; \theta, \mu) = 1/[1 + \exp\{-\theta(x - \mu)\}]$. Their criterion for Bayesian D-optimality is to maximize the average log determinant of the normalized information matrix; the expectation is taken according to a prior distribution on the coefficients (μ, θ). Their method requires that the number of design points be specified, and so they repeated the optimization using the simplex algorithm of Nelder and Mead (1965) starting with 2 design points and increasing the number steadily up to 20 points. They then chose the design that optimized the criterion on the smallest number of design points. They illustrated their method with μ and θ uniformly distributed on an interval and with three different intervals for each parameter.

Chaloner and Larntz (1989) demonstrated that as the uncertainty increases, so does the minimum number of support points required to attain the optimal value. However, in a 1987 technical report, they also showed that out of three intervals examined for μ, only for the widest interval, when it is distributed uniformly on $[-1, 1]$, is the Bayesian design significantly more efficient than the best local D-optimal design. A design based on clustering yields similar results and has three support points for the examples where the Bayesian design has three support points. It is more interesting to evaluate the effectiveness of a design based on clustering for the examples in which the Bayesian design proved superior to the centroid local D-optimal design, that is, for $\mu \sim \mathrm{U}[-1, 1]$. As discussed by Chaloner and Larntz (1989), the choice of interval for θ has only a small influence on the final design and its efficiency, and we display the results when $\theta \sim \mathrm{U}[6, 8]$. Their optimal Bayesian design uses seven support points with a reported value of -4.5783 for the average log of the information matrix determinant.

We used K-means clustering over 100 local D-optimal designs corresponding to coefficients of θ and μ set by a Niederreiter (1988) quasi-random sequence over the described intervals. Similar to Chaloner and Larntz (1989), we increased the value of K, the number of support points, from 2 to 20. Figure 3 shows the mean value of the log of the determinant matrix when estimated using the same 100 local designs. Similar to the reported result, the criterion seems to reach a stable value for a design with seven support points, and that value seems to be better than the one stated by Chaloner and Larntz.

Averaging over 100 designs may be insufficient for a precise evaluation, and using the same coefficient values to create the cluster and to estimate its performance may create a bias. Therefore, we reevaluated the 7-support point design (created through 100 local D-optimal designs) using 10,000 local D-optimal designs, with their coefficients again determined by a Niederreiter sequence. The criterion value given by this more thorough evaluation confirmed the validity of the rough estimation. Its value is -4.25, higher than the value reported for the Bayesian design.

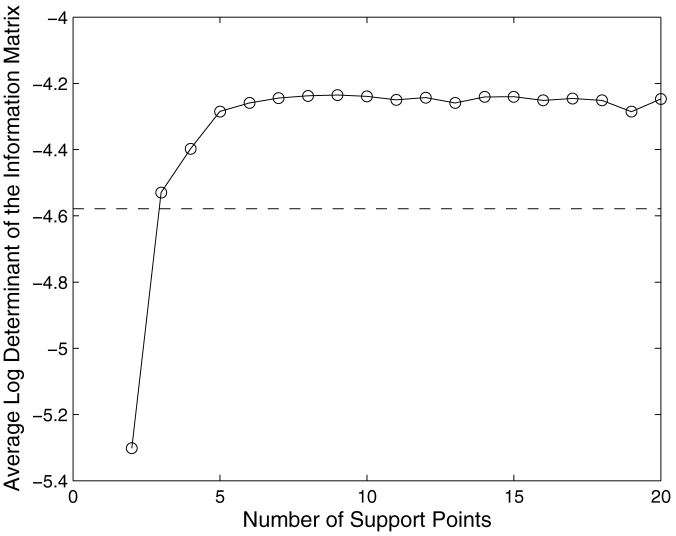

Figure 3. Mean Value of the Log of the Determinant Matrix Estimated Over a Rough Grid. The dotted line represents the Chaloner and Larntz (1989) reported value.

One of the advances of the work of Chaloner and Larntz (1989) over previous approaches is to create designs without the requirement that the points be equally spaced, and with the possibility of a different number of observations at each point. Like their work, a design created by clustering is not restricted to equally spaced points, but it does put equal weight on all of the support points. For a given set of points, it is possible to improve the design using sequential quadratic programming to adjust the weights. For the given example, this leads to only a minor improvement in the criterion value, to -4.23.

Even though creating a robust design using clustering was found to be superior in this example, Bayesian designs would be expected to be generally better. If clustering normally does not fall much from Bayesian designs, then it has clear advantages over them: simplicity of creation and the need for considerably less computational resources. Unlike Bayesian design, extending the clustering procedure to multivariate problems is almost trivial and is considered next.

4. CLUSTERING VERSUS MULTIVARIATE COMPROMISE DESIGNS

Woods et al. (2006) provided a method for finding exact designs for experiments in which there are several explanatory variables. They used simulated annealing to find (as in Chaloner and Larntz 1989) a design with a given number of support points that maximizes the average log determinant of the normalized information matrix. They noted that evaluating the integral is too computationally intensive for incorporation within a search algorithm, and thus they averaged over a partial set chosen to represent the model space. Their method allows creation of compromise designs with uncertainty in the link function, the linear predictor, and the model parameters.

Woods et al. (2006, sec. 5) gave an example of creating a 16-point compromise design across a parameter space. They described a crystallography experiment aimed at modeling how four explanatory variables (rate of agitation during mixing, volume of composition, temperature, and evaporation rate) affect the probability that a new product is formed. They recommended that when the suggested ranges for the unknowns, β_i, are not large, the local D-optimal design for the centroid of the parameter space will be used. Otherwise, a compromise design based on a coverage design performs better. The superiority of the compromise design created using a coverage set is demonstrated with a parameter space as described in Table 1, which is based on parameter space $\mathcal{B}_3$ in table 1 of the original article.

A design's performance was evaluated using the median and minimum efficiency relative to 10,000 local D-optimal designs created for random parameter vectors from the parameter space. The efficiency of a design was calculated as $(|M_C|/|M_L|)^{1/p}$, where p is the number of unknown coefficients and M_C and M_L are the information matrices for the evaluated and local D-optimal designs. A standard 2^4 factorial design performed poorly for the example with a median efficiency value of .07 and a minimum of .003.

Before creating a design using clustering, we examined the compatibility of our assessments to those given by Woods et al. (2006). We created 10,000 local D-optimal designs using the procedure described in Section 2. The values of the 10,000 parameter vectors were produced by a base 2 Niederreiter quasi-random sequence with 2^{12} as a seed. This procedure enables recreation of the exact parameter vectors used here and at the same time promises more uniform coverage of the parameter space than can be achieved by random sampling. We then used these designs to evaluate the median and minimum efficiency of the coverage design of Woods et al. (2006). The results were compatible with those reported by Woods et al.: a median of .415 (vs. .41), and a minimum of .113 (vs. .12).

The procedure of Woods et al. (2006) requires their special algorithm and a good problem-dependent choice of numerous tuning parameters, and is computer-intensive. The current timing for their algorithm (see *www.maths.soton.ac.uk/staff/woods/glm_design*; Woods 2006) is roughly 1.5 minutes on a stronger computer than we used, and 7 minutes on our computer. We proceeded in an attempt to create an alternative design by clustering.

First, we created local D-optimal designs for 100 parameter values, continuing the Niederreiter quasi-random sequence used so far to ensure the use of different locally optimal designs for creating the composite design and assessing its efficiency. This preparation work took less than 1 minute. We then gathered the 1,600 resulting points and applied K-means clustering, as implemented in MATLAB (MathWorks Inc.) to choose 16 representative points as our design. Often optimal design points are found on the boundary of the design region; thus we used the sum of the absolute differences as a distance measure, so that each cluster is represented by the componentwise median of its points.

Table 1. Coefficient Ranges From the Woods et al. (2006) Crystallography Experiment

Parameter	*Range*
β_0	[−3, 3]
β_1	[4, 10]
β_2	[5, 11]
β_3	[−6, 0]
β_4	[−2.5, 3.5]

Each time that clustering is performed, a slightly different design emerges. This is due to the random choice of initial cluster centroid positions. Thus we summarize the design performance through the median (and minimum) efficiencies averaged over 50 identical clustering runs, using the notation *mean* [95% CI], where "CI" denotes confidence interval.

Clustering was found to have results comparable to those of the Woods et al. (2006) composite design, with median efficiency of .40 [.38, .42] and minimum efficiency of .091 [.06, .12]. The time it took to create the composite design was negligible: .25 seconds [.16, .33], in addition to the 1-minute preparation phase for finding 100 local D-optimal designs.

Better results can be obtained by repeating the clustering process numerous times. Similar to Chaloner and Larntz (1989) and Woods et al. (2006), we chose the cluster with the highest average log determinant of the information matrix. Averaging was done on the rough grid of 100 parameter vectors used to create the local D-optimal designs. Indeed, repeated clustering improved the results; the median efficiency grew to .423 [.416, .430], and the minimum efficiency was .096 [.06, .13], requiring only 25 seconds to choose the design.

Furthermore, because clustering is very fast, we can easily examine the effect of different choices for the number of support points. Figure 4 displays the result of clustering done with different numbers of support points. For each number of support points, we used clustering only once, based on the 100 local D-optimal designs. We approximated the efficiency using the same locally optimal designs. Given the local designs, producing the data for the figure took only 20 seconds.

From Figure 4, we see that the median efficiency reaches a stable value around or slightly above the previous number of 16 support points, but the minimum efficiency continues to grow and stabilizes for only 30 support points or more. This

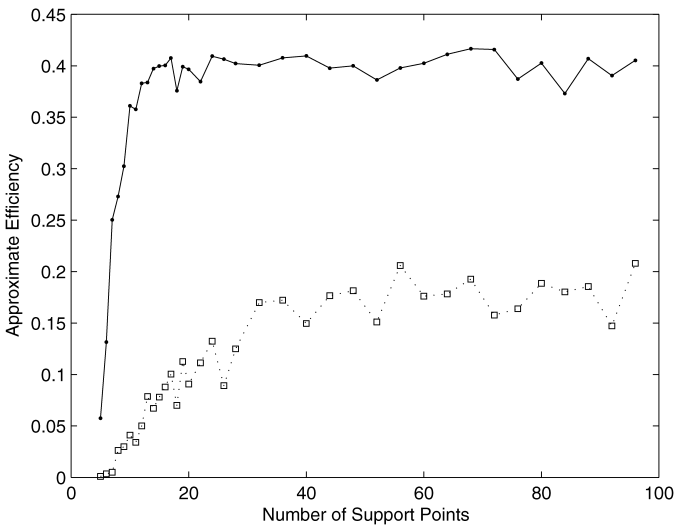

Figure 4. The Effect of Different Choices for the Number of Support Points on the Approximated Efficiency (—•— median efficiency; ···□··· minimum efficiency).

shows that a design with more support points may be advised. In fact, Woods et al. (2006) stated that in the crystallography experiment, 48 observations will be run, with the 16-point design replicated 3 times. Woods (2006) reported that computation time for a 24-point design is about twice as long as that needed for their 16-point composite design, which may be why they used replicates rather than considering the option of adding new support points. Creating a 48-point design using the algorithm of Woods et al. (2006) required more than 100 minutes on our computer.

Given this, we chose, as before, the best design out of 100 repetitions for a 48-means clustering. As expected, the median did not change much: .423 [.415, .432], but the minimum efficiency increased to .177 [.141, .213]. The increase in the minimal value is of great importance, as we discuss in Section 6. In addition, it is found that efficiency estimation based on 100 local D-optimal designs is quite accurate, so one can produce both a compromise design and an estimate of its efficiency distribution based on a small sample of local designs that is easy to obtain.

Producing the 48-point design that exceeds the efficiency of the results of Woods et al. (2006), requires only an additional 72 [63, 80] seconds, and, combined with the preparation phase, takes only 2.5 minutes, compared with the 100 minutes needed to generate a 48-point design with the procedure of Woods et al. (2006) on our computer. The proposed method is simpler, both conceptually and computationally. Not only is it faster and of higher efficiency, but, due to the production of local D-optimal design, it facilitates the possibility of evaluating the efficiency of the design and comparing the efficiency of designs with different numbers of support points. Without this unique trait, an algorithm produces a design with no measure of its effectiveness, and the importance of using a 48-point design might not have been recognized.

5. ROBUSTNESS FOR LINEAR PREDICTORS AND LINK FUNCTIONS

The method of Woods et al. (2006) for finding compromise designs allows uncertainty not only in the model parameters, but also in the link function and the choice of the linear predictor. In section 6 of their article they give an example with two explanatory variables in which there is uncertainty as to whether a first-order model or a model with the interaction term is more appropriate, and also uncertainty about the link function: probit versus the asymmetric complementary log–log (CLL). The values of the model parameters were $\boldsymbol{\beta} = (3.0, 1.6, 4.1)'$ for the first-order model and $\boldsymbol{\beta} = (1.2, 1.7, 5.4, -1.7)'$ when considering a model with the interaction term. The results given are for designs with six observations.

Woods et al. (2006) showed that for this example, all of the four locally optimal designs perform poorly for some of the possible characteristics, with the first-order local D-optimal designs insufficient for any estimation of the interaction term. A compromise design created for the same problem enables estimation of all four models with efficiency of at least .64. Table 2 is a reproduction of table 3 of Woods et al. (2006), adding a column with the efficiency achieved by clustering the four local D-optimal designs.

Table 2. Efficiencies of a Design Produced by Clustering, the Woods et al. (2006) Compromise Design, and Four Local Optimal Designs d_i Reproduced From Table 3 of the Original Article

		Design					
Model		*Clustering*	*Woods*	d_3	d_4	d_5	d_6
Probit	No interaction	.75	.77	1.00	.34	.99	.30
	Interaction	.81	.80	0	1.00	0	.97
CLL	No interaction	.64	.64	.99	.24	1.00	.11
	Interaction	.85	.86	0	.97	0	1.00

It is seen that the performance of the Woods et al. (2006) compromise design and the design created by clustering the local D-optimal designs is very similar; both achieve at least moderate efficiency for all four models.

Besides demonstrating the heuristic qualities of clustering, this example is useful for demonstrating limitations of its use. Three of the local D-optimal designs included a replicate of the point $[1, -1]$, and so had only five support points for a six-point design. This poses an obstacle for clustering, because although the best design may put higher weight on this support point than on the other design points, the output of the clustering procedure includes any single point only once, and if seeking a six-point design, it is likely to replace the replication of the existing point by adding a different point with inferior contribution. To overcome this obstacle, we slightly jittered the points of the local D-optimal designs. Indeed, clustering the jittered design points puts two points very close to $[1, -1]$ and is an easy way to overcome the limitation.

6. WAVE SOLDERING EXAMPLE

Wu and Hamada (2000, chap. 13, p. 563) discussed a two-level factorial experiment to study the number of defects in a wave soldering process. We now consider how our approach could be applied to such an experiment, focusing on the number of solder-joint defects and five continuous process variables: prebake temperature, flux density, conveyer speed, cooling time, and solder temperature. Like Wu and Hamada (2000), we use a Poisson model for the defect counts.

The first step in our algorithm is to define an a priori distribution for the model parameters. Here we consider the common situation when no previous experimental data are available for this task. As an alternative, we suggest constructing the prior distribution in collaboration with an expert from the factory. The expert is asked to estimate the number of defects for different possible settings of the five factors; his estimates are analyzed as if they were experimental results, and the uncertainty modeled is presented to the expert for approval. We consider a case in which the expert believes that a first-order model would be sufficient, but also specifies two pairs of factors that have possible interaction effects. Both a first-order model and a model with two cross-product terms were constructed from the analysis of the thought experiment. The estimations are presented in Table 3, for factor levels coded to $[-1, 1]$.

Remark 1. Note that the standard errors (SEs) are much larger for the model that contains interactions, even though both

Table 3. Prior Coefficient Estimates for Two Models for the Wave Soldering Example

	First-order		With interactions	
Term	*Estimate*	*SE*	*Estimate*	*SE*
Intercept	−1.52	.21	−2.35	.69
x_1	−4.30	.20	−5.53	.94
x_2	−1.79	.16	−2.99	.82
x_3	−3.39	.24	−3.95	.59
x_4	−.28	.32	−.86	.54
x_5	.23	.30	.41	.36
x_1x_2			−2.07	1.32
x_1x_3			−1.13	.98

NOTE: SE, standard error.

models were estimated from the same data. This phenomenon of having less precise estimates for more complex models is common.

Remark 2. Our analysis did not take into account the correlation of coefficients, but this can be easily addressed, if desired, in sampling the parameter vectors used to generate local D-optimal designs.

Efficiency was estimated using 20,000 local D-optimal designs, half for the first-order model and the other half for a model with the suspected interactions. For each model, local D-optimal designs were found for 10,000 coefficient vectors sampled from the normal distribution through a quasi-random sequence.

6.1 Clustering versus a Full Factorial Design

The median efficiency of a full factorial experiment with 32 points is $<.1$ and, as shown in Figure 5, its distribution has 2 peaks, originating from the 2 models considered. (The full factorial has higher efficiency for the first-order model.) The efficiency can be greatly improved using clustering. As a preparation phase, we created a set of 200 local D-optimal designs, 100 for each model, with parameters taken from a quasi-random sequence in accordance with the normal distribution assumed.

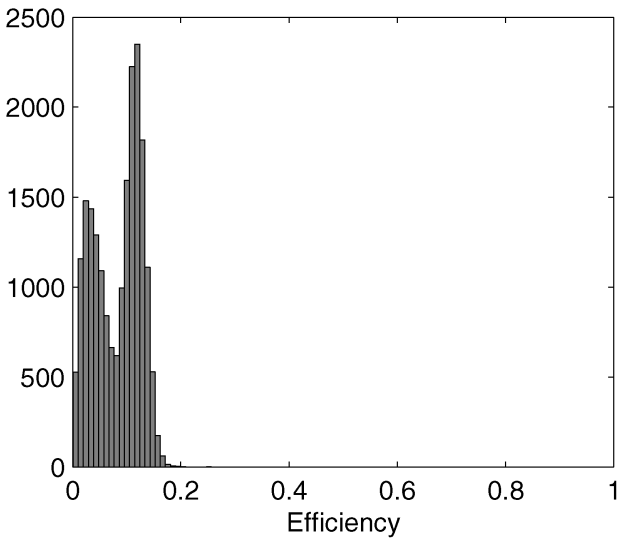

Figure 5. A Full Factorial Design Efficiencies Histogram for 2 Considered Models With 10,000 Representative Model Parameters Each.

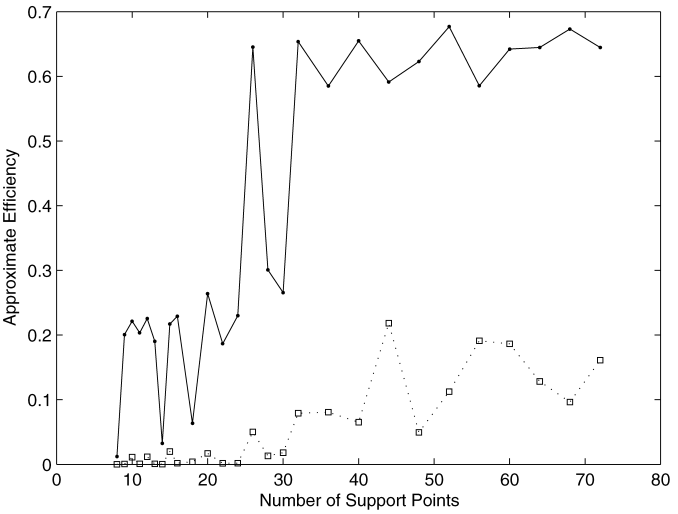

Figure 6. The Effect of Different Choices for the Number of Support Points on the Approximated Efficiency (—•— median efficiency; ·□· minimum efficiency).

The next step was to choose a good number of support points. We repeated the process used with the crystallography experiment, clustering only once for each of a set of possible support point numbers, and evaluating the efficiency only roughly over the same set of parameter vectors.

Remark 3. For our purposes, it is sufficient to cluster only once for any tested number of support points without any repetitions, as was done in the production of Figure 6. But lack of repetition causes some of the cases studied to perform very poorly, due to a bad random choice of the initial K cluster centroids when performing the K-means clustering procedure. Hence the graph is not smooth, and the "dips" observed at around 15 and 30 support points are likely to be an effect of a poor clustering solution related to the random initial choice of centroids, not to a real problem with these design sizes. Using this graph, we choose the desired value for K; then it is important to repeat the clustering process numerous times, to ensure high efficiency.

Remark 4. It should be assumed that when the unknown parameters' uncertainty is distributed normally, the true minimum efficiency should approach 0. Thus the values of the lower curve in Figure 6 are not representative for the minimum values. But we argue that the lower curve is still a good indicator for the expected change in small efficiency quantiles.

It is seen that the median efficiency is stable for any choice of more than 30 support points. If we now choose, for example, a design with 48 support points, then the median efficiency (as evaluated with the comprehensive database of 20,000 local D-optimal designs) is .65. Figure 7 displays a histogram of local efficiencies for a 48-run design achieved by clustering.

6.2 Clustering versus Centroid Design

As noted by Woods et al. (2006), the local D-optimal design for the centroid of the parameter space is often a sufficiently robust design. When there is more than one model, as in our example, there is no single centroid. Still, given a strong relationship between the two examined models, one of the two

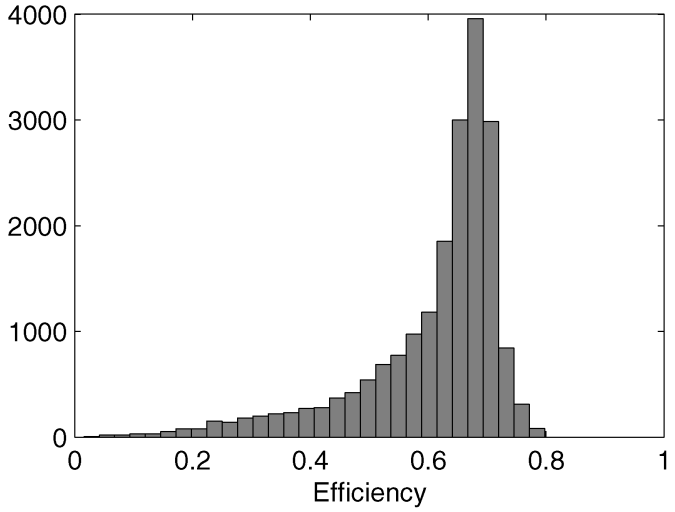

Figure 7. Efficiencies Histogram for a 48-Point Cluster Design for 2 Considered Models With 10,000 Representative Model Parameters Each.

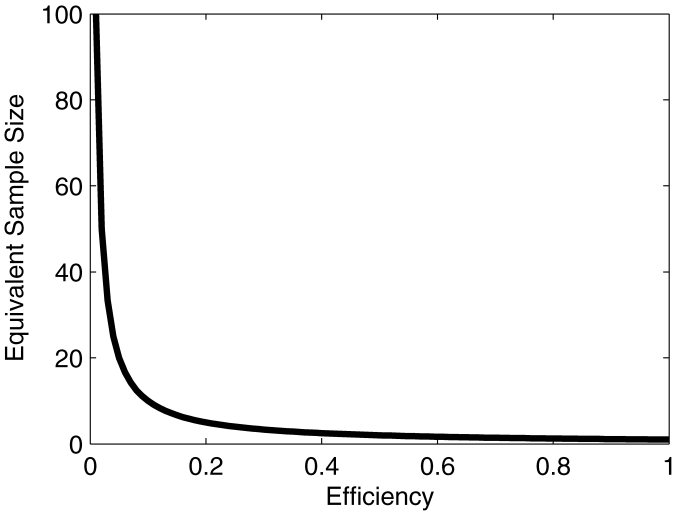

Figure 9. The Importance of Having as Small a Portion as Possible of Low Efficiencies.

centroids may be a good choice. Indeed, the local D-optimal design for the richer model is found to perform well, as shown by its efficiency histogram in Figure 8.

The centroid design's median efficiency is even higher in this case than the efficiency achieved by clustering: .69. Furthermore, being a local optimal design, the histogram is guaranteed to reach a maximum efficiency of 1. As a result, it may seem that a different example would better demonstrate the advantages of creating designs by clustering; possible examples include experiments with more models being considered (perhaps with a greater distinction between them) or a wider uncertainty in the parameter space, as is often the case when the expert cannot give one set of estimates, but considers different scenarios.

Even in this example, however, the design created by clustering has an advantage over the centroid design, hidden in the left region of the histograms. The relative efficiency between any two designs can be considered an equivalent sample size; if the relative efficiency of one design is ρ, then it requires $1/\rho$ times as many observations to achieve the same D-criterion value as the design with which it was compared. As is visually obvious (Fig. 9), an efficiency value $<.2$ is related to a drastic increase in the required sample size. It is much more important for a robust design to have as small a fraction of low efficiencies as possible, rather than including high efficiencies.

Comparison of the efficiency histograms of the centroid and cluster designs shows that the left tail of the cluster design is thinner. In fact, for the cluster design, only 2% of the 20,000 models considered have efficiency $<.2$, compared with 4.5% of the models for the centroid design. Hence clustering creates a more robust design by decreasing the portion of the uncertainty space that, if discovered to be the true setup, would make the design seriously inefficient.

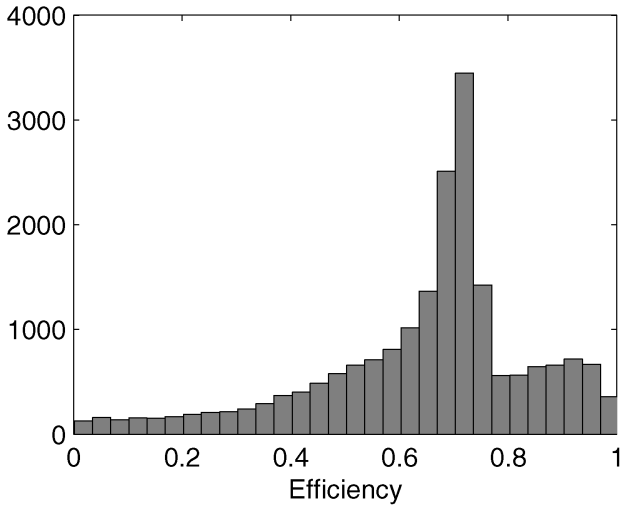

Figure 8. Centroid Local D-Optimal Design Efficiencies Histogram.

7. ALGORITHM SUMMARY

We now summarize the algorithm for creating of a robust design through clustering:

1. Translate previous experimental results or experts' opinion into a set of possible models, with a clear statement of the uncertainty as to needed terms and coefficient values.
2. For each model, linear predictor, link function, and/or target criterion, create a sequence of possible parameter vectors according to a defined distribution, as agreed on in the first step. Sampling the parameter space using a low-discrepancy sequence should be preferred over a random sample. In the examples provided, we used 100 vectors produced by a Niederreiter (1988) low-discrepancy sequence.
3. Find locally optimal designs for all of the sequences created in step 2; see Section 2 for details.
4. Group the local designs from all models into a single matrix. Apply slight jittering on the components; we decreased from the absolute value of each matrix element a uniformly distributed random variate on $[0, 10^{-4}]$.
5. Choose a number of support points, K, and use a K-means clustering procedure on the matrix to produce a design. We recommend using the sum of the absolute differences as a distance measure, so that each centroid will be the componentwise median of the points in each cluster. In

MATLAB, this can be done using the "kmeans" function with the option "cityblock" for distance.

6. Repeat the process for various choices of K, to choose the most appropriate value.
7. For the chosen K, apply clustering numerous times; we used 100 repetitions. After each clustering attempt, calculate the information matrix of the outcome for all of the models and parameter vectors chosen in step 2. Sum the log of the determinants of the information matrices. Use the clustering output with the highest sum as the design.

8. CONCLUSION

Local D-optimal designs for GLMs can be easily found using existing algorithms and computer packages with minor adjustments. Creating a database of locally optimal designs in accordance with an a priori formulation of uncertainty of the model in the parameter space, the model considered, link function, and so on can be used to find a design robust to all aspects of the described uncertainty. The proposed heuristic is then to cluster the resulting database. Clearly, this is a simple procedure, requiring minimal computational resources or time even for complex models.

Our algorithm benefits from a unique trait, having the ability to evaluate the design's efficiency through the database of locally optimal designs used in generating the robust design. This unique trait, together with the speed of the process, allows exploration of various designs and investigation of the effect of choosing different numbers of support points is encouraged. Special attention should be given to finding designs with as small a fraction of very low efficiencies as possible, say, $<.2$. It has been demonstrated that the ability to explore many alternative designs in a short time helps this simple procedure outperform more sophisticated and complex design-optimization methods.

ACKNOWLEDGMENTS

The authors thank Dave Woods, University of Southampton, U.K., for providing helpful comments, engaging in productive discussions, and sharing his design code.

APPENDIX: LOW–DISCREPANCY SEQUENCES

Here we provide background on low-discrepancy sequences in general and particularly on Niedrreiter's (1988) quasi-random sequence. Source code for an implementation for MATLAB, C++, and Fortran90 can be found at *http://www.csit.fsu.edu/~burkardt/m_src/niederreiter2/niederreiter2.html*. In addition to providing the source code, the site briefly explains the nature of the algorithm:

"A quasirandom or low discrepancy sequence, such as the Faure, Halton, Hammersley, Niederreiter or Sobol sequences, is 'less random' than a pseudorandom number sequence, but more useful for such tasks as approximation of integrals in higher dimensions, and in global optimization. This is because low discrepancy sequences tend to sample space 'more uniformly' than random numbers. Algorithms that use such sequences may have superior convergence."

We used NIEDERREITER2, which, as explained in the foregoing, is an adaptation of the INLO2 and GOLO2 routines in ACM TOMS Algorithm 738. The original code can compute only the "next" element of the sequence. The revised code allows the user to specify the index of any desired element. The original, true, correct version of ACM TOMS Algorithm 738 is available in the TOMS subdirectory of the NETLIB website.

A.1 An Illustration

Figure A.1 compares 100 pseudorandom observations on $[0, 1]^3$, produced by the command "RANDOM = rand(100, 3)" in MATLAB to a three-dimensional Niederreiter base 2 low-discrepancy sequence with 2^{12} used as a seed, produced with the code suggested above. The upper row of the figure contains the two-dimensional projections of the pseudorandom sequence, and the bottom row presents the corresponding projections for Niederreiter's quasi-random sequence. Clearly, the low-discrepancy sequence covers the space more evenly, avoiding the empty gaps that are common in the pseudorandom sequence.

A brief overview on the mathematical foundations of low-discrepancy sequences can be found in Wikipedia, The Free Encyclopedia, at *http://en.wikipedia.org/w/index.php?title=Low-discrepancy_sequence&oldid=27681750*; the rest of this appendix is a part of the description in the quoted link.

A low-discrepancy sequence is a sequence with the property that for all N, the subsequence $x_1, \ldots, x_N$ is almost uniformly distributed (in a sense to be made precise), and $x_1, \ldots, x_{N+1}$ is almost uniformly distributed as well. Low-discrepancy sequences are also called quasi-random or subrandom sequences because of their use in situations similar to those when pseudo-random or random numbers are used instead. The "quasi"-modifier is used to denote more clearly that the numbers are not random (and to differentiate them from pseudorandomness, which uses different assumptions), but have useful properties similar to randomness in certain applications, such as the quasi-Monte Carlo method.

The notion of uniformity is made precise as the discrepancy defined later. Roughly speaking, the discrepancy of a sequence is low if the number of points falling into a set B is close to the number that would be expected from the measure of B. At least three methods of numerical integration can be phrased as follows: Given a set $x_1, \ldots, x_N$ in the interval $[0, 1]$, approximate the integral of a function f as the average of the function evaluated at these points,

$$\int_0^1 f(u)\,du \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i).$$

If the points are chosen as $x_i = i/N$, then this is the rectangle rule. If the points are chosen to be randomly (or pseudorandomly) distributed, then this is the Monte Carlo method. If the points are chosen as elements of a low-discrepancy sequence, then this is the quasi-Monte Carlo method. A remarkable result, the Koksma–Hlawka inequality, shows that the error of such a method can be bounded by the product of two terms, one of which depends only on f and another that is the discrepancy of the set $x_1, \ldots, x_N$.

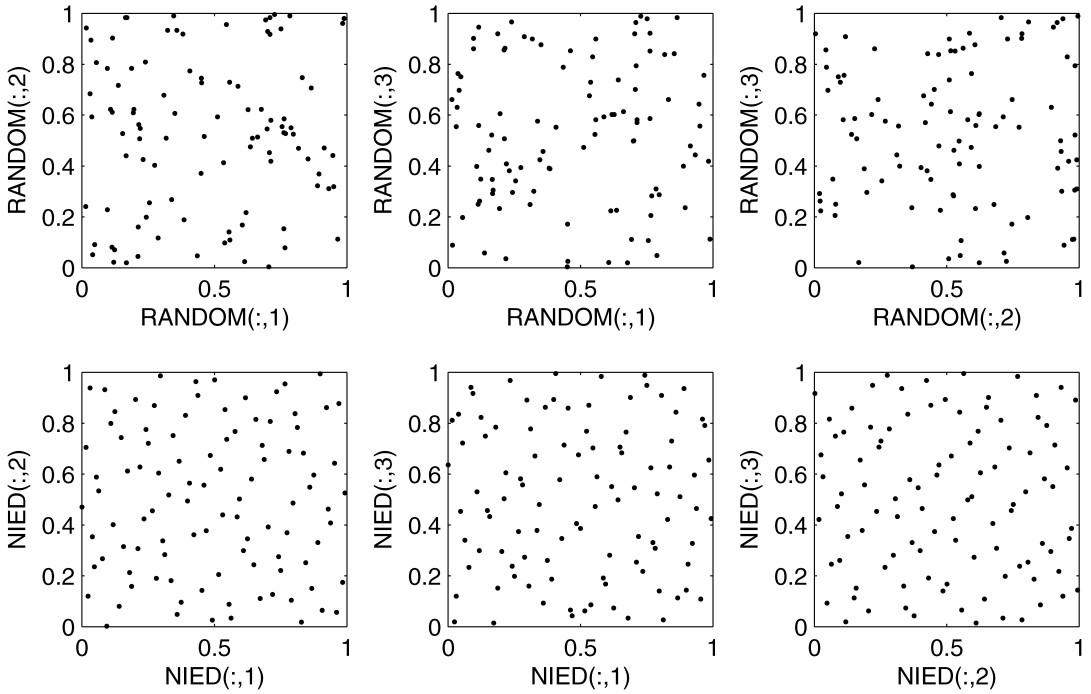

Figure A.1. Comparison of Two-Dimensional Projections of a Pseudorandom Sequence (top row) and Niederreiter's Quasi-Random Sequence (bottom row).

It is convenient to construct the set $x_1, \ldots, x_N$ in such a way that if a set with $N + 1$ elements is constructed, then the previous N elements need not be recomputed. The rectangle rule uses a point set that has low discrepancy, but in general the elements must be recomputed if N is increased. Elements need not be recomputed in the Monte Carlo method if N is increased, but the point sets do not have minimal discrepancy. Using low-discrepancy sequences gives the quasi-Monte Carlo method the desirable features of the other two methods.

A.2 Definition of Discrepancy

The star-discrepancy is defined as follows, using Niederreiter's notation:

$$D_N^*(P) = \sup_{B \in J^*} \left| \frac{A(B; P)}{N} - \lambda_s(B) \right|,$$

where P is the set $x_1, \ldots, x_N$, λ_s is the s-dimensional Lebesgue measure, $A(B; P)$ is the number of points in P that fall into B, and J^* is the collection of sets of the form $\prod_{i=1}^{s} [0, u_i)$, where u_i is in the half-open interval $[0, 1)$. Therefore, $D_N^*(P) = \| \text{disc} \|_\infty$, where the discrepancy function is defined by $\text{disc}(y) = A([0, y); P)/N - \lambda_s([0, y))$.

A.3 Two Main Conjectures

Conjecture 1. There is a constant c_s depending only on s, such that $D_N^*(x_1, \ldots, x_N) \geq c_s (\ln N)^{s-1}/N$ for any finite point set $x_1, \ldots, x_N$.

Conjecture 2. There is a constant c_s' depending only on s, such that $D_N^*(x_1, \ldots, x_N) \geq c_s' (\ln N)^s/N$ for any infinite sequence $x_1, x_2, x_3, \ldots$.

These conjectures are equivalent; they have been proved for $s \leq 2$ by W. M. Schmidt. In higher dimensions, the corresponding problem remains open. The best-known lower bounds are due to K. F. Roth.

A.4 The Best-Known Sequences

Constructions of sequences are known (due to Faure, Halton, Hammersley, Sobol', Niederreiter, and Van der Corput) such that $D_N^*(x_1, \ldots, x_N) \leq C(\ln N)^s/N$, where C is a certain constant, depending on the sequence. After Conjecture 2, these sequences are believed to have the best possible order of convergence.

[Received February 2006. Revised June 2006.]

REFERENCES

Abdelbasit, K. M., and Plackett, R. L. (1983), "Experimental Designs for Binary Data," *Journal of the American Statistical Association*, 78, 90–98.

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

Chaloner, K., and Larntz, K. (1989), "Optimal Bayesian Design Applied to Logistic Regression Experiments," *Journal of Statistical Planning and Inference*, 21, 191–208.

Federov, V. V. (1972), *Theory of Optimal Experiments*, New York: Academic Press.

Ford, I., Torsney, B., and Wu, C. F. J. (1992), "The Use of a Canonical Form in the Construction of Locally Optimal Designs for Non-Linear Problems," *Journal of the Royal Statistical Society*, Ser B, 54, 569–583.

Hardin, R. H., and Sloane, N. J. A. (1993), "A New Approach to the Construction of Optimal Designs," *Journal of Statistical Planning and Inference*, 37, 339–369.

Hedayat, A. S., Yan, B., and Pezzuto, J. M. (1997), "Modeling and Identifying Optimum Designs for Fitting Dose-Response Curves Based on Raw Optical Density Data," *Journal of the American Statistical Association*, 92, 1132–1140.

Khuri, A. I., Mukherjee, B., Sinha, B., and Ghosh, M. (2004), "Design Issues for Generalized Linear Models," Technical Report 2004-016, University of Florida, Dept. of Statistics.

MacQueen, J. B. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA: University of California Press, pp. 281–297.

Mathew, T., and Sinha, B. K. (2001), "Optimal Designs for Binary Data Under Logistic Regression," *Journal of Statistical Planning and Inference*, 93, 295–307.

Nelder, J. A., and Mead, R. (1965), "A Simplex Method for Function Minimization," *Computer Journal*, 7, 308–313.

Niederreiter, H. (1988), "Low-Discrepancy and Low-Dispersion Sequences," *Journal of Number Theory*, 30, 51–70.

Robinson, K. A., and Khuri, A. I. (2003), "Quantile Dispersion Graphs for Evaluating and Comparing Designs for Logistic Regression Models," *Computational Statistics and Data Analysis*, 43, 47–62.

Sitter, R. R. (1992), "Robust Designs for Binary Data," *Biometrics*, 48, 1145–1155.

Sitter, R. R., and Torsney, B. (1995a), "Optimal Designs for Binary Response Experiments With Two Design Variables," *Statistica Sinica*, 5, 405–419.

——— (1995b), "D-Optimal Designs for Generalized Linear Models," in *MODA4—Advances in Model-Oriented Data Analysis (Refereed Proceedings of the 4th Conference on Model-Oriented Data Analysis)*, C. P. Kitsos and W. G. Muller, eds., Spetses, Greece: Physica-Verlag, Heidelberg, pp. 87–102.

Woods, D. C. (2006), "Designing Experiments for Binary Data via Simulated Annealing," Technical Report 391, University of Southampton, School of Mathematics.

Woods, D. C., Lewis, S. M., Eccleston, J. A., and Russell, K. G. (2006), "Designs for Generalized Linear Models With Several Variables and Model Uncertainty," *Technometrics*, 48, 284–292.

Wu, C. F. J., and Hamada, M. (2000), *Experiments: Planning, Analysis, and Parameter Design Optimization*, New York: Wiley.