Key Words: Computer Experiments, Space-filling Experimental Designs, Nonrectangular Experimental Regions

Abstract

Maximin distance designs are space-filling experimental designs that were introduced by Johnson, Moore, and Ylvisaker (1990). Unlike Latin hypercube and orthogonal array designs, the definition of maximin designs does not depend on the geometry of the design space; hence, maximin designs are potentially attractive for experiments that involve nonrectangular design spaces. Unfortunately, maximin designs are difficult to compute. To address this difficulty, we consider a framework in which approximations to maximin designs can be obtained as solutions to nonlinear programming problems in which the constraints define the design space. In this framework, our ability to calculate designs corresponds to our ability to manage the optimization constraints. Finally, we discuss some computational issues that arise when attempting to calculate designs in this manner.

1 Introduction

Many deterministic computer simulations are extremely expensive to evaluate. For example, Booker (1996) described a simulation of the aeroelastic and dynamic response of a helicopter rotor blade for which a single function evaluation requires approximately six hours of cpu time on a Cray Y-MP. In such situations, a common engineering practice (Barthelemy and Haftka, 1993) is to replace the expensive simulation, which we represent as a real-valued function $f$, with an inexpensive surrogate, $\hat{f}$. The surrogate is constructed from information obtained by evaluating $f$ at a set of carefully selected design sites. This report considers the problem of how to choose a specified number of design sites in a sensible way.

The problem of choosing design sites is a problem of experimental design. This observation, together with the observation that various techniques from spatial statistics can be gainfully employed to construct the surrogates, led statisticians to begin to study the design and analysis of computer experiments. Surveys of the rapidly growing literature on this subject have been made by Sacks, Welch, Mitchell, and Wynn (1989), by Booker (1994), and by Koehler and Owen (1996).

Roughly speaking, there are two approaches to the design of computer experiments. The parametric approach is often supplied with a Bayesian interpretation. One assumes that $f$ is the realization of a stochastic process and specifies a parametric family of possible processes. It is possible to extend familiar design criteria like D-optimality to this setting, then construct designs that are optimal with respect to the specified family. However, the practical difficulties of actually computing such optimal designs can be formidable.

The nonparametric approach to the design of computer experiments chooses design sites in a manner that is perceived to be "space-filling". When the experimental region $E \subseteq \mathbb{R}^p$ is a bounded rectangle, Latin hypercube sampling (McKay, Beckman, and Conover, 1979) and orthogonal array sampling (Owen, 1992, 1994; Tang, 1993) are practical, inexpensive ways of generating space-filling designs. Design criteria that are explicitly space-filling include the minimax and maximin distance principles proposed by Johnson, Moore, and Ylvisaker (1990); however, these designs can be difficult to compute. The purpose of this report is to describe a method of approximating maximin designs that exploits conventional nonlinear programming algorithms.

The utility of Latin hypercube and orthogonal array sampling diminishes when, as is often the case, the experimental region is not rectangular. If $E$ can be inscribed in a rectangle of the same dimension, then a plausible space-filling design can often be obtained by the simple ad hoc device of generating a space-filling design for the circumscribing rectangle and accepting the resulting design sites that fall in $E$. 

\footnote{Supported in part by AFOSR Grant F49620-95-1-0210, by the Institute for Computer Applications in Science and Engineering (ICASE) under NASA Contract NAS1-19480, and by a gift from the Mobil Technology Company.}
Our desire to improve such designs—and to generate plausible space-filling designs in regions that do not readily lend themselves to this device—motivates the work described in this report.

2 An Exact Formulation

Let \( N \) denote the specified number of design sites and suppose that \( x_1, \ldots, x_N \in E \), where \( E \) is a compact subset of \( \mathbb{R}^p \). For convenience, we place \( x_i \) in row \( i \) of the \( N \times p \) design matrix \( X = (x_{ik}) \). We then abuse notation and write \( X \in E \).

Let

\[
d_{ij}(X) = \|x_i - x_j\|_2 = \left( \sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^{1/2}.
\]

Then \( X^* \in E \) is a maximin Euclidean distance design in \( E \) if and only if

\[
\min_{i < j} d_{ij}(X^*) \geq \min_{i < j} d_{ij}(X)
\]

for all \( X \in E \), i.e., if and only if \( X^* \) is a global solution of the nonsmooth optimization problem

\[
\begin{align*}
\text{maximize} & \quad \min_{i < j} d_{ij}(X) \\
\text{subject to} & \quad X \in E.
\end{align*}
\]

(1)

Maximin designs are intuitively appealing because they explicitly endeavor to spread the design sites as much as possible. Other metrics are certainly possible; in this report, we restrict attention to Euclidean distance.

In general, explicit formulae for \( X^* \) will not exist and maximin designs must be computed numerically. To do so, it may be helpful to reformulate Problem (1) as a smooth nonlinear programming problem. Imagine positioning each site in a prospective design \( X \) at the center of a closed ball of radius \( r \), where \( r \) is small enough that the intersection of any pair of balls is either empty or a singleton point. To spread the sites in accordance with the maximin distance criterion, we seek a design for which \( r \) can be made as large as possible. Thus, \( X^* \) is a maximin Euclidean distance design in \( E \) if and only if it solves

\[
\begin{align*}
\text{maximize} & \quad r \\
\text{subject to} & \quad X \in E, d_{ij}(X) \geq 2r;
\end{align*}
\]

or, equivalently, if and only if \( X^* \) solves

\[
\begin{align*}
\text{minimize} & \quad -r^2 \\
\text{subject to} & \quad X \in E, [d_{ij}(X)]^2 \geq 4r^2, r \geq 0.
\end{align*}
\]

(2)

There are interesting analogies between the problem of maximin distance design and certain problems in signal detection. Very recently, Gockenbach and Kearsley (1999) considered the problem of designing optimal sets of signals (finite time series) under amplitude constraints and in the presence of non-Gaussian noise. They considered a signal set optimal when the largest probability of mistaking any one signal for any other is minimal. This is a constrained minimax problem that the authors reformulated as a smooth nonlinear program that is strikingly similar to Problem (2).

Gockenbach and Kearsley (1999) noted that their nonlinear program has several unfortunate features, shared by our Problem (2), that typically confound standard nonlinear programming algorithms:

“Difficulties arise when one tries to solve... with standard algorithms. The fact that there are far fewer variables than constraints results in three specific difficulties:

- There are ‘almost’ binding constraints at the solution.
- The linearized constraints are often inconsistent.
- The boundary of the feasible region is noticeably nonlinear.”

Nevertheless, the authors were able to find solutions using a sequential quadratic programming algorithm that incorporates a constraint perturbation technique studied by Kearsley (1996). It seems likely that these methods can be adapted for finding maximin distance designs via Problem (2); however, this is a topic for future research. The purpose of the present report is to examine approximate formulations of the maximin distance design problem that can be solved by standard algorithms for nonlinear programming.

3 Approximate Formulations

Let \( \phi \) denote any strictly decreasing function on \((0, \infty)\), e.g., \( \phi(t) = 1/t \), and let \( \phi_{ij}(X) = \phi(d_{ij}(X)) \). Let \( v(X) \) denote the vector of length \( N(N-1)/2 \) whose \( k \)th component is \( \phi_{ij}(X) \), where

\[
k = (j - 1)(N - j/2) + i - j
\]

for \( j = 1, \ldots, N - 1 \) and \( i = j + 1, \ldots, N \). Then \( X^* \) is a maximin design if and only if it is a global solution of the constrained minimax problem

\[
\begin{align*}
\text{minimize} & \quad \|v(X)\|_{\infty} \\
\text{subject to} & \quad X \in E.
\end{align*}
\]

(3)
We approximate $\|v(X)\|_\infty$ by the smooth objective function $\|v(X)\|_\sigma$, resulting in the more tractable optimization problem

$$\begin{align*}
\text{minimize} & \quad \|v(X)\|_\sigma \\
\text{subject to} & \quad X \in E.
\end{align*}$$

(4)

With $\phi(t) = 1/t$, this formulation was exploited by Morris and Mitchell (1995) in their study of distance designs that are maximin within the class of Latin hypercube designs.

Let $X^\sigma$ denote a global solution of Problem (4). The following result justifies calling $X^\sigma$ an approximate maximin design, although we submit that the plausibility of $X^\sigma$ as a space-filling design does not depend on this justification.

**Theorem 1** Let $\sigma_k \to \infty$ as $k \to \infty$ and let $X^\infty$ be any accumulation point of $\{X^\sigma\}$. Then $X^\infty$ is a maximin distance design.

**Proof:** For every $\sigma \in [1, \infty]$,\n
$$\|v(X^\sigma)\|_\sigma = \|v(X^\infty)\|_\sigma$$

as $k \to \infty$. Hence, by diagonalization,

$$\|v(X^\sigma)\|_\sigma \to \|v(X^\infty)\|_\infty$$

as $k \to \infty$.

Let $X^\sigma$ denote any global solution of Problem (3). If $X^\infty$ is not a global solution of Problem (3), then there exists $\epsilon > 0$ such that

$$\|v(X^\infty)\|_\infty \geq \|v(X^\sigma)\|_\infty + 2\epsilon.$$

Choose $K$ large enough that $k \geq K$ entails both

$$\|v(X^\sigma)\|_\sigma - \|v(X^\infty)\|_\infty < \epsilon$$

and

$$\|v(X^\sigma)\|_\sigma - \|v(X^\sigma)\|_\infty < \epsilon.$$

Then

$$\|v(X^\sigma)\|_\sigma < \|v(X^\sigma)\|_\sigma,$$

which is a contradiction. \qed

Once $\sigma$ has been fixed, Problem (4) is equivalent to the following:

$$\begin{align*}
\text{minimize} & \quad \sum_{j>l}[\phi_{ij}(X)]^\sigma \\
\text{subject to} & \quad X \in E.
\end{align*}$$

(5)

How difficult it will be to solve Problem (5) will depend on how easily one can manage (a) the objective function, which in turn will depend on the choices of $\phi$ and $\sigma$; and (b) the constraints.

It is not always easy to choose $\phi$ and $\sigma$ in such a way that Problem (5) is sensibly scaled. Because the scaling of Problem (5) will affect the performance of the algorithms that we use to solve, we endeavor to ameliorate this situation by assuming that $E$ has been scaled so that $E \subset [0,1]^p$.

The transformation $\phi$ should be constructed in such a way that $\phi(t)$ is large when $t$ is small. If $\phi$ has a singularity at $t = 0$, then the optimization algorithm will be strongly discouraged from considering designs in which sites are replicated. Thus, $\phi(t) = 1/t$ is a natural transformation. So is the computationally less expensive $\phi(t) = 1/t^2$. We have experimented with several transformations and currently prefer

$$\phi(t) = \log \left(1 + \frac{1}{t^2}\right).$$

(6)

For $\phi(t) = 1/t$, Morris and Mitchell (1995) tried $\sigma = 1, 2, 5, 10, 20, 50, 100$ until they were convinced that their simulated annealing algorithm had found a maximin design. Our goal is more modest, in that we are content to settle for a reasonable approximation to a maximin design. For $\phi(t) = 1/t^2$ and (6), we have found that $\sigma \in [5, 10]$ usually results in satisfactory designs.

Morris and Mitchell (1995) noted the following tradeoff: as $\sigma$ increases, $X^\sigma$ better approximates a maximin design, but Problem (5) is more difficult to solve. We have observed the same phenomenon, but generally we have been impressed with the performance of standard nonlinear programming algorithms on Problem (5). In contrast to the global search strategy employed by Morris and Mitchell (1995), we have only studied local search strategies. These strategies are sometimes trapped by nonglobal minimizers and do not always find $X^\sigma$. Nevertheless, for reasonable choices of $\sigma$, they consistently find designs whose minimal intersite distances are almost as great as those realized by $X^\sigma$.

There remains the matter of the constraints that define the experimental region, $E$. In order to treat Problem (5) as a nonlinear program, we assume that $E$ is defined by finite numbers of equality and (more typically) inequality constraints and we hope that there are not too many highly nonlinear constraints. A plethora of nonlinear programming algorithms are available; the reader seeking to identify algorithms and software suited to specific applications is referred to Moré and Wright (1993). To take advantage of powerful computing platforms and state-of-the-art software with a minimal investment of time and effort, we recommend submitting jobs written in AMPL (A Mathematical Programming Language).
documented by Fourer, Gay, and Kernighan (1993), to the NEOS (Network-Enabled Optimization System) server at Argonne National Laboratories:

http://www.mcs.anl.gov/neos/Server

Currently, our preferred solver is SNOPT.

4 An Example

We conclude by demonstrating an approximate maximin Euclidean distance design that comprises \( N = 16 \) sites in a simple nonrectangular region \( E \subset [0,1]^2 \). The decision variables are the \( Np = 32 \) elements of

\[
X = [x_{ik}],
\]

which of course must satisfy the bound constraints

\[
0 \leq x_{ik} \leq 1.
\]

To define \( E \), we imposed summation constraints, requiring that

\[
0.6 \leq x_{i1} + x_{i2} \leq 1.0.
\]

We constructed an initial design \( X^0 \in E \) by pseudorandom sampling from a uniform distribution on \( E \). This was accomplished by using the S-Plus function \texttt{runif} to generate pseudorandom observations from a uniform distribution on \([0,1]^2\) and accepting those in \( E \) until \( N = 16 \) feasible sites had been generated. The initial design \( X^0 \) is displayed in Figure 1. Note that

\[
\min_{i<j} d_{ij}(X^0) \doteq 0.03,
\]

Problem (5), with \( \phi \) specified by (6), \( \sigma = 10 \), and \( X = X^0 \), was coded in AMPL and submitted (by email to neos@mcs.anl.gov) to the SNOPT solver via NEOS. The text of this submission can be obtained from the author (trosset@math.wm.edu). The SNOPT solver took just 0.65 seconds to compute \( X^\sigma \), the approximate maximin Euclidean distance design displayed in Figure 2. Note that

\[
\min_{i<j} d_{ij}(X^\sigma) \doteq 0.18,
\]

so that \( X^\sigma \) represents a six-fold improvement on \( X^0 \) with respect to the maximin distance design criterion.

Although much testing remains to be done, especially with more sites in higher-dimensional experimental regions defined by more challenging constraints, we believe that this example provides a proof of concept. Given access to the appropriate algorithms and software developed by the numerical optimization community, approximate maximin distance designs offer considerable promise for constructing space-filling designs in nonrectangular regions.

![Figure 1: A Pseudorandom Sample from a Uniform Distribution on E](image1)

![Figure 2: An Approximate Maximin Euclidean Distance Design in E](image2)
Acknowledgments

I am grateful for suggestions made by Andrew Booker and Max Morris, and for preliminary computational experiments performed by Laura Taylor.

References


