A New Approach to the Construction of Optimal Designs

R. H. Hardin
N. J. A. Sloane

AT&T Bell Laboratories
Murray Hill, New Jersey 07974 USA

September 28, 1992

ABSTRACT

By combining a modified version of Hooke and Jeeves' pattern search with exact or Monte Carlo moment calculations, it is possible to find I-, D- and A-optimal (or nearly optimal) designs for a wide range of response-surface problems. The algorithm routinely handles problems involving the minimization of functions of 1000 variables, and so for example can construct designs for a full quadratic response-surface depending on 12 continuous process variables. The algorithm handles continuous or discrete variables, linear equality or inequality constraints, and a response surface that is any low degree polynomial. The design may be required to include a specified set of points, so a sequence of designs can be obtained, each optimal given that the earlier runs have been made. The modeling region need not coincide with the measurement region. The algorithm has been implemented in a program called `gosset` , which has been used to compute extensive tables of designs. Many of these are more efficient than the best designs previously known.

A New Approach to the Construction of Optimal Designs

R. H. Hardin
N. J. A. Sloane

AT&T Bell Laboratories
Murray Hill, New Jersey 07974 USA

1. Introduction

This paper presents a new algorithm for constructing experimental designs. The algorithm, described in Section 3, has the following features.

a. The variables may be discrete or continuous (or both — none of these choices are mutually exclusive), discrete variables may be numeric or symbolic, continuous variables may range over a cube or a ball, and the variables may be required to satisfy linear equality or inequality constraints (so mixtures and constrained mixtures can be handled).

b. The model to be fitted may be any low degree polynomial (for example a quadratic response-surface).

c. The number of runs is specified by the user (so minimal designs present no difficulty).

d. The design may be required to include a specified set of points (so a sequence of designs can be found, each of which is optimal given that the earlier measurements have been made).

e. The region where the model is to be fitted need not be the same as the region where measurements are to be made. (So for example designs can be optimized for modeling over a continuous region even if the measurements are constrained to a discrete set. Designs for extrapolation can be obtained in a similar way.)

f. The algorithm is capable of minimizing any differentiable optimality criterion. We have focused on four such criteria, I-, A-, D- and E-optimality, with a distinct preference for the first, which minimizes the average prediction variance (see Section 2). The I-optimality criterion requires knowledge of the moments of the design region, and the algorithm finds these moments either from the exact formulae or by Monte Carlo estimation.

g. The user has control over how much effort is expended by the algorithm, and can if desired monitor the progress of the search. It is not necessary to specify initial points for the search.

h. The algorithm can be used in situations in which the errors, rather than being independent, have a known correlation matrix.

The algorithm appears to be powerful enough to find optimal or nearly optimal designs involving as many as 1000 continuous variables, for example a full quadratic response-surface design depending on 12 process variables. Its effectiveness decreases as the number of discrete variables increases. However, the algorithm has found D-optimal main-effect designs involving 20 2-level factors (i.e. with 420 discrete variables).

No individual ingredient of our algorithm is new. But although computers have been used by many people to construct experimental designs (surveys of such work can be found for example in [13, Chap. 15], [21], [27], [70], [78], [79], [87], [93], [99]), we believe that no algorithm comparable to ours is presently available.

Implementation. We have implemented the algorithm in the C language in a program called `gosset`. A built-in parser permits a very flexible input. A brief description is given in Section 4, and further details can be found in the users' manual [50].

The program is named after the amateur mathematician *Thorold Gosset* (1869–1962), who was one of the first to study polytopes in six, seven and eight dimensions [23, p. 164], and his contemporary, the statistician *William Seally Gosset* (1876–1937), who was one of the first to use statistical methods in the planning and interpretation of agricultural experiments [82]. Although from our geometric viewpoint their work is related, we do not know if the paths of Thorold (Cambridge, London, lawyer), and William Sealy (Oxford, Dublin, brewer) ever crossed.

Applications. So far there have been two main uses for `gosset` .

(i) We have attempted to construct optimal (especially I-optimal) designs for a number of "classical" situations, for example linear, quadratic or cubic response-surface designs with k continuous variables in a cube or ball with n runs, over quite a large range of values of k and n, typically $1 \leq k \leq 12$ and n ranging from the minimal value to 6 (or more) greater than the minimal value. We have also computed designs for similar models and regions in which the variables are discrete. An extensive library of these designs is now built into `gosset` . Our work on these "classical" problems can be regarded as an attempt to provide optimal "exact" designs with small numbers of runs to complement the "asymptotic" designs of Kiefer et al. [38], [39]–[47],[63]–[68].

As we shall see in Section 5, some of these designs overlap with and improve on designs already available in the literature. Other designs we have found will be published elsewhere (see [51], [52], [53], [54]).

We have also used this collection of designs as data for theoretical investigations. Two results are worth mentioning here.

(a) There is a simple lower bound on the average variance of an I-optimal design for quadratic models with n measurements in a k-dimensional ball (see Eq. (13) below). A number of interesting designs meet the bound. There is a similar bound for D-optimal designs.

(b) It is known that for large numbers of runs D- and G-optimal designs are equivalent [65]. The results of Section 5 show that I-optimal designs are strictly different. For example, I-optimal designs make more measurements at the center of the region and fewer at the boundary. For quadratic models in a k-dimensional ball, k large, an I-optimal design makes about $4/k^2$ of the measurements at the center of the sphere, compared with about $2/k^2$ for D- and G-optimal designs. I-optimality also appears to be a more strict condition than D-optimality. In situations where the criteria produce similar designs (such as certain linear designs, see Section 5.3), we commonly find that although I-optimal designs are D-optimal, the converse is not necessarily true.

(ii) We have constructed designs for a number of industrial applications, for instance problem involving continuous and discrete variables simultaneously, with linear constraints on the variables (see Section 5.4).

We must emphasize that the theoretical justification for our designs (see Section 2) depends strongly on the validity of the particular model being used. We are assuming that the investigator has carried out some exploratory investigations (for example a screening design) and has identified a region where it is plausible to describe the response by a low degree polynomial. (Of course the design points when the initial measurements have been made can be incorporated by our program in the next design — see Section 4.)

This work began in 1990 when a statistician in Seattle, David H. Doehlert, wrote to one of us asking if we could construct designs for a full quadratic response-surface depending on k

variables in a sphere, where k is between 3 and 14, and in which the number of runs is minimal or close to minimal. We had been using the pattern search algorithm in studying the Tammes problem of placing M points on a k-dimensional sphere so they are well-separated [55], and we found that a similar approach could be used for constructing designs.

2. Choice of optimality criterion

Extensive discussions of the relative merits of A-, D-, E-, G- and I-optimality and of the dangers of relying on any single numerical criterion have appeared in the literature (see for example [13], [15], [48], [64], [87]), and our treatment will be brief.

Suppose for concreteness that we wish to construct a design for a full quadratic response-surface model

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j + \epsilon \ , \tag{1}$$

where there are k variables $x_1, \ldots, x_k$, $p = (k+1)(k+2)/2$ unknown coefficients β, and the errors ϵ are independent with mean 0 and variance σ^2. Let the design consist of $n \geq p$ points

$$[x_{j1}, \ldots, x_{jk}] \quad \text{for} \quad 1 \leq j \leq n \ , \tag{2}$$

chosen from a certain *region of measurement* (or *operability*) O. Let X be the $n \times p$ expanded design matrix, containing one row

$$f(x) = [1, x_1, \ldots, x_k, x_1^2, \ldots, x_k^2, x_1 x_2, \ldots, x_{k-1} x_k]$$

for each design point $x = [x_1, \ldots, x_k]$, and let

$$M_X = \frac{1}{n} X' X$$

denote the matrix of moments of the design measure (the prime indicates matrix transposition). The prediction variance at an arbitrary point x is

$$\operatorname{var} \widehat{y}(x) = \frac{\sigma^2}{n} f(x) M_X^{-1} f(x)' \ .$$

We define an *I-optimal* design (following Box and Draper [11], [12] and others) to be one which minimizes the *normalized average* or *integrated prediction variance*

$$I = \frac{n}{\sigma^2} \int_R \operatorname{var} \widehat{y}(x) d\mu(x) \ , \tag{3}$$

where R is the *region of interest* (or *modeling region*), and μ is uniform measure on R with total measure 1. This integral simplifies (cf. [12], p. 341) to give

$$I = \mathrm{trace}\{MM_X^{-1}\} \ , \tag{4}$$

where

$$M = \int_R f(x)' f(x) d\mu(x) \tag{5}$$

is the moment matrix of the region of interest.

In contrast, A-, D-, E- and G-optimal designs are those which minimize

$$\begin{aligned} A &= \text{trace } M_X^{-1} \ , & (6)\\ D &= \{\det \ M_X\}^{-1/p} \ , & (7)\\ E &= \text{maximal eigenvalue of } M_X^{-1} \ , & (8)\\ G &= \text{maximal value of var } \widehat{f}(x), \quad x \in R & (9) \end{aligned}$$

respectively. We shall refer to the quantities defined in (4), (6)–(9) as the I-, A-, etc. *values* of the design. As the number of runs $n \to \infty$, these quantities approach limits I_∞, A_∞, etc., and the I-, A-, etc. *efficiencies* of the design are given by

$$\frac{I_\infty}{I}, \ \frac{A_\infty}{A}, \ \frac{D_\infty}{D}, \ \frac{E_\infty}{E}, \ \frac{G_\infty}{G} \tag{10}$$

(cf. [2], [70]).

We are most interested in minimizing the prediction variance, var $\widehat{f}(x)$, for $x \in R$, which suggests the use of I- or G-optimality. On the other hand our algorithm requires that the criterion be differentiable, which holds for A-, D- and I- but not E- or G-optimality. Furthermore we wish to be able to construct designs for which the measurement region O and modeling region R are distinct, which also picks out the I- or G-criteria (since the A-, D- and E-criteria do not involve R). We have therefore chosen I-optimality as our primary criterion. However, the implementation of the algorithm allows the user to search for any of I-, A-, D- or E-optimal designs.

It is worth mentioning that the I-value (Eq. (4)) is a dimensionless quantity. Also, provided the model has the property that if it contains one monomial of degree d then it contains all possible monomials of degree $\leq d$, then the I-value is unchanged if the variables are rescaled or if an orthogonal transformation is applied to the variables. In contrast the A- and E-values of a design depend on the particular choice of coordinate axes used, a property which seems unsatisfactory. In the present paper we shall concentrate on I- and D-efficiencies. For the ball or cube it is not difficult to determine I_∞ and D_∞ exactly. In other cases we determined them empirically, by estimating the limiting values of I and D as the number of runs increases. We attempted to calculate I_∞ and D_∞ to at least three decimal places of accuracy. However, inaccuracies in I_∞ and D_∞ are not too significant, because they do not change the relative efficiencies of comparable designs. Note that a design is I- (or D-) optimal if it has the highest I- (or D-) efficiency for a given number of runs, and so a design can be optimal without being 100% efficient.

To guard against the dangers of using any single number as a design criterion, we have plotted *variance dispersion graphs* (Giovannitti-Jensen and Myers [48], Vining [95]) for our designs; some examples will be found in [51]. These graphs show how the prediction variance var $\widehat{f}(x)$ varies over the region of interest, and also give the G-value of the design.

We conclude this section by briefly mentioning some earlier papers that look for I-optimal designs.

Studden [94] gives a method for constructing approximations to I-optimal designs for 1-dimensional problems.

Haines [49], Meyer and Nachtsheim [72], Crary [24] and Snow and Crary [92] all use simulated annealing to search for I-optimal designs. This approach seems to be restricted to low dimensions and small numbers of design points, and furthermore is less successful at finding optimal designs than our approach. For example, `gosset` finds slightly different designs in two of the four cases shown in Figure 1 of [24]. Meyer and Nachtsheim [72] give several quite small examples where simulated annealing was unable to find the best designs known. One of these, a 17-run four-dimensional example, is described in Section 5.2 below.

Crary's [24] program, I-OPT, like ours, allows the measurement and modeling regions to be distinct. I-OPT has a feature that `gosset` does not have at present, namely the ability to specify relative weights for points in the modeling region.

3. The algorithm

For a given model (such as (1)) involving k variables and p unknown coefficients, a design consists of n points (2) (not necessarily distinct) in the region O. Let $\mathbf{x}$ be a vector specifying the coordinates of *all* the design points. We wish to choose the design vector $\mathbf{x}$ so as to minimize a certain differentiable function $F(\mathbf{x})$. For I-optimal designs, F is the normalized average prediction variance $I = \text{trace } MM_X^{-1}$, where M is the moment matrix of the region R (see (4), (5)), while for A- or D-optimality F is trace M_X^{-1} or $\{\det\ M_X\}^{-1/p}$ (see (6), (7)). We let $\mathbf{g}(\mathbf{x})$ denote the gradient $\nabla F(\mathbf{x})$, normalized to have length $\sqrt{n}$.

The regions O and R can be quite complicated. The present implementation (see Section 4) permits O and R to be a product of cubes, balls and finite sets, possibly intersected by hyperplanes and half-spaces. This includes simplices, of course, since these are intersections of cubes and hyperplanes.

If there are no equality constraints in the problem then the design vector $\mathbf{x}$ is simply a concatenation of the individual points (2) of the design. When equality constraints are present we choose new, internal, variables to parametrize the space O, and then $\mathbf{x}$ is a concatenation of the coordinates of the internal variables.

If the region O is a simple connected space such as a ball, our basic strategy is to minimize F using a modified version of the Hooke and Jeeves [59] *pattern search*. An excellent description is given by Beightler et al. [7], and our discussion here will be brief. We have modified the algorithm to make use of the gradient of F. Roughly speaking, pattern search minimizes F by accelerating down the gradient, and setting the velocity to zero whenever the function does not improve. The velocity increases as long as improvements continue.

We begin with an initial design vector $\mathbf{x} = \mathbf{x}^{(0)}$, with velocity $\mathbf{v}^{(0)} = 0$. This initial design may be obtained by choosing n random points from O, although in the implementation the user has the option of specifying $\mathbf{x}^{(0)}$ himself. We attempt to define the next vector by

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \mathbf{v}^{(i+1)} , \tag{11}$$

where the velocity vector $\mathbf{v}^{(i+1)}$ is given by

$$\mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} - s\,\mathbf{g}(\mathbf{x}^{(i)}) , \tag{12}$$

for $i \ge 0$, where s is the current step size. If $F(\mathbf{x}^{(i+1)}) < F(\mathbf{x}^{(i)})$ the step is a success, we accept this value for $\mathbf{x}^{(i+1)}$, and repeat the iteration with s multiplied by 1.04. Otherwise we set $\mathbf{v}^{(i)} = \mathbf{0}$ and try (11) and (12) again. If there is still no reduction in F, we divide the step size by 2, and if it is larger than some small accuracy limit, try (11) and (12) again. Otherwise we terminate this attempt, pick another starting design $\mathbf{x}^{(0)}$, and repeat the whole procedure. Any particular sequence $F(\mathbf{x}^{(0)}), F(\mathbf{x}^{(1)}), F(\mathbf{x}^{(2)}), \ldots$ (a decreasing, hence bounded, sequence of positive numbers) is terminated either when the improvement is less than some small accuracy limit, or (very rarely) when the number of steps exceeds a specified limit. After a specified number of attempts the algorithm terminates, returning the best and the most recent second-best designs found.

Equations (11) and (12) must be modified when the design vector $\mathbf{x}^{(n)}$ involves points near the boundary of O (either the natural boundary, for instance when O is a ball, cube or simplex, or a boundary imposed by a constraint on the variables). When this happens, the components of the gradient vector crossing the boundary are simply zeroed before normalization, and if any point in $\mathbf{x}^{(n)}$ nevertheless moves outside the boundary we move it to the closest point on the boundary.

Incidentally, in the implementation the gradient is calculated from the exact formula for ∇F, rather than by numerical differentiation of F.

We have found this method very robust and efficient, typically minimizing a function of N variables in N steps.

The above procedure is used when all or most of the variables in the design are continuous. When many discrete variables are present we use a collection of heuristic algorithms (which do however work most of the time).

Suppose for concreteness that there are $k = 6$ variables, the first four of which are continuous and the last two are discrete variables. Let the n points of the design be

$$\mathbf{a} = a_1 \cdots a_6, \quad \mathbf{b} = b_1 \cdots b_6, \ \ldots, \quad \mathbf{c} = c_1 \cdots c_6 \ ,$$

so that the design vector is $\mathbf{x} = \mathbf{ab} \cdots \mathbf{c}$.

We first discuss the question of choosing the discrete coordinates a_5, a_6, b_5, b_6, $\ldots$, c_5, c_6 of the initial design vector $\mathbf{x}^{(0)}$. For this we use one (or more) of three strategies: (i) run sequentially through all possibilities in some fixed order, running through them all once before repeating any; (ii) choose them at random from the set of all discrete possibilities; (iii) choose them at random from the underlying continuous space. If constraints are present, in all three cases only points satisfying the constraints are used. Strategies (i) and (ii) are reasonable when the number of combinations of discrete variables is not much larger than the number of design points, otherwise (iii) appears to be more successful. In the implementation the user can specify the strategy to be used, or accept the default strategy (which depends on the type of problem).

Second, in the optimization step, we optimize (using the method described earlier in this section) either just the continuous components of $\mathbf{x}$, keeping the discrete components fixed, or all components, treating the discrete components as continuous.

If the latter strategy is adopted, a final step is needed to convert the discrete components (a_5, a_6, etc.) back to their permitted discrete values. For this discretization step we again use one of three methods, replacing each continuous component by: the closest discrete value, whichever of the two closest discrete values gives the smallest value of F, or whichever of the discrete values gives the smallest value of F. We have found the second of the three methods works best, the third being too slow. All three methods go through the components of $\mathbf{x}$ in order, making a decision and not changing it again. These are "greedy" techniques, and do not try to avoid getting trapped at local minima. Again the user can, if desired, control which method is used.

There is a complication to the discretization step, however, for the resulting vector $\mathbf{x}$ may now violate some constraints involving the discrete variables. If so, we attempt to adjust both the continuous and discrete components of $\mathbf{x}$ to correct this. Again we use one of three methods, the most successful of which is the following. We first find the closest vector $\mathbf{x}'$ (say) to $\mathbf{x}$ that satisfies all the constraints, treating the discrete components as continuous again. Then we generate a number of random legal vectors that satisfy all the constraints and have continuous or discrete components as appropriate, and pick the ten of these that are closest to $\mathbf{x}'$. In computing their distance to $\mathbf{x}'$, we only consider components that are involved in constraints (the others being irrelevant here). From these ten vectors we select the one that minimizes F (again using only the components that are involved in constraints). This vector is properly discrete and satisfies all the constraints. A final optimization is then made with respect to the continuous variables only.

The modeling region, R, enters the algorithm only through the moment matrix M (see (5)). For simple regions (balls, cubes, discrete sets) we use the exact moment matrix (cf. [11], [12]), otherwise we estimate them by a Monte Carlo method. In the present implementation the most general region R is a subset of a region Ω (say) which is a product of balls, cubes and discrete sets. The algorithm generates random points with a uniform distribution over Ω. If the point satisfies all the constraints it is accepted and its contributions towards the entries of M are recorded. The user has the option of controlling the number of Monte Carlo samples used.

The more samples used, the more accurate is the matrix. It is usually not too serious for nontheoretical applications if the matrix is inaccurate — this corresponds to a slight change in shape of the region of modeling.

Two features of the algorithm are worth mentioning. If the user discovers that greater accuracy in the moment matrix is called for, further Monte Carlo trials can be run, and the results will be automatically combined with the previous estimates. Second, the algorithm checks to see if the region O is invariant under changing the sign of any coordinates or under permutation of the coordinates and, if so, automatically makes M invariant under the same symmetries.

4. Implementation: the program `gosset`

The algorithm has been implemented in the `C` language in a program called `gosset` . The program takes about 10,000 lines of code, runs on a variety of UNIX©-based machines, and is distributed in conjunction with the `S` statistical programming language [6]. Only a brief description is given here; full details can be found in the manual [50].

There are three basic steps in using `gosset` : specifying the design, calculating the moment matrix, and searching for the optimal design.

Step 1. Design specification. The design is specified by a series of numbered lines, somewhat like a `BASIC` program. Variables are specified by lines such as

```
10 range a b c 90 120
20 sphere Temp Zinc Water center 300 20 4 radius .5
30 discrete KEV 70 90 100
```

which indicate that a, b, c are continuous variables ranging from 90 to 120; Temp, Zinc and Water belong to a ball centered at [300, 20, 50] of radius .5, and KEV is a quantitative discrete variable with values 70, 90, 100. Any number of such lines may appear.

To allow a more flexible input format, the program includes its own parser. (Programs such as `YACC` and `LEX` (see [62]) have made people forget how easy it is to write a parser starting from scratch.)

Lines 10–30 above are contravariant specifications of scaling, but it is often more convenient to use a covariant form. Line 20 for example could equally well be written

```
20 sphere 2*(Temp-300) 2*(Zinc-20) 2*(Water-4)
```

since the default radius is 1 and the default center is the origin. `Gosset` accepts either form, or any combination of them.

Unprimed variables describe the region of measurement (O), and matching primed variables describe the region of modeling (R). The absence of primed variables indicates that O and R coincide. For example

```
10 discrete a b c -1 1
20 range a′ b′ c′ -1 1
```

specifies that a, b, c are 2-level discrete variables taking the values -1 and 1, while the model is to be fitted over the whole cube $[-1, 1]^3$. If line 20 is omitted, the model would be fitted just on the finite set $\{-1, 1\}^3$ where the measurements are to be made. In this example the numbers -1 and 1 could be omitted, because by default all variables have limits -1 and 1, and discrete variables are 2-valued.

Constraints (which must be linear) are expressed by lines such as

```
50 constraint x+2*y+3*z=1
60 constraint A<B B+0.4<C
```

The specification `use` forces a given point to appear in the design (for example, runs made in an earlier experiment, the results of which we do not want to lose). The program will then search for the best design that includes these points. For example

```
70 use x=.9 y=.3 material="plastic"
```

Inequality constraints may be violated by `use` points. It may be one of these that suggested the wisdom of the constraint! (However, because the program handles equality constraints by eliminating variables, `use` points must satisfy all equality constraints.)

The model is specified symbolically, by an expression such as

```
80  model  (1+a+b+c)^2-c^2
```

which specifies that the model (see Eq. (1) above) contains the terms 1, a, b, c, a^2, b^2, ab, ac, bc, but not c^2. In the model specification, $+$ means "add to the list", $-$ means "take away if it exists", and multiplication means "form all products and keep one representative of each type of term". This is similar to the way models are specified in the `S` language [18].

Once the design has been specified by a "program" in this way, the program is compiled by the command

```
compile
```

At this point gosset chooses new, internal, variables to parametrize the spaces R and O, eliminating each equality constraint by dropping a variable, whose range is then expressed by a pair of inequalities, eliminating clearly redundant or unnecessary inequalities, and scaling the internal variables so they are in the range -1 to 1.

Step 2. Moment calculation. The second step is to find the moment matrix M (see previous section) by Monte Carlo estimation with a command such as

```
moments   time=300   n=10^6
```

which would sample until either 300 seconds have elapsed or 10^6 samples have been taken. If this is a problem where the program can determine M exactly, the effect will be as if the user had said

```
moments n=0
```

and the exact moments will be used. The `moments` command makes a header file moments.h which is used by the search programs.

Step 3. Searching for an optimal design. The third step is to search for an optimal design, using a command such as

```
design type=I runs=24 n=200
```

This would instruct the program to search for an I-optimal design with 24 runs, taking 200 random designs, optimizing each of them (using the techniques described in Section 3), and choosing the best.

The `design` command has many options. For example, `type=I` (or `D`, or `A`) searches for an I- (or D- or A-) optimal design. `Extra=3` searches for a design with three more runs than a minimal design would contain (and relieves the user of the necessity of counting the terms in the model!). `Start=v.16.old` begins the search at a previous design. `Tiny=1.0e-5`, `steps=500`, `time=300` set limits to the minimal step size, the number of steps in the search, and the search time. `Processors=7`, on a multiprocessor machine, would run seven searches in parallel (with of course a dramatic increase in speed). In the `design` command, the user can, if desired, control the search algorithm by the `program=` option. `Program=r` means choose a random start, **v** minimizes I by changing the continuous variables while keeping the discrete variables fixed, `d` is similar but maximizes det M_X, `vc` minimizes I by changing all variables, treating discrete variables as if they were continuous, and so on.

These symbols can be combined using various operators: `+` means choose the best of the left and right sides, `^` means iterate the left side the number of times indicated by the right side, and `*` is a "pipe" command which passes a design on to the next process.

For example

```
design program=(r*v)^50+lib
```

means generate 50 random starts, optimize each, take the best, and compare with the best design found in the library.

If the `program=` option is not used, `gosset` uses various default programs, for example `program=r*v` if there are no discrete variables.

`Gosset` automatically builds up a library of designs for each problem as it proceeds, and, as mentioned in Section 5, contains a very extensive built-in library of designs for the ball, cube, etc.

`Gosset` does many things simultaneously — reads input from the terminal, spins off jobs to run the Monte Carlo moment calculation, to compute the I-value of a design, etc., and also checks from time to time to see if these jobs are finished. Several difficulties had to be overcome in order to do this in a portable and convenient way.

For example, one cannot directly use the UNIX `wait` system call to see if a job is finished without being blocked (put to sleep until a termination) if the job is still unfinished. Of course this would prevent `gosset` from responding to the terminal. So every `wait` system call in `gosset` is preceded by spinning off a dummy job that does nothing. Then the `wait` is sure to complete quickly, because the dummy job terminates quickly, and if the job we are really interested in is finished we will be told about that too.

However, some systems always report completed jobs in the reverse order, reporting the most recent jobs first. So `gosset` runs a test when it starts up, and if it discovers that jobs are reported in the reverse order, makes the dummy jobs sleep for one second (instead of doing nothing), so they are certain to be reported after other completed jobs.

`Gosset` also contains its own "input daemon" to make it possible to read input without halting the main program. The input daemon program simply waits until input lines appear, and then writes them to a disk file. The main program does not read input directly (this would halt the main program if no input were present), but checks occasionally to see if the disk file has grown, and if so, reads the last line there.

These features make it possible for a user to monitor the progress of the **moments** or **design** programs, using **watch**, **status** or **kill** commands, to inspect the best design found so far, or to perform other tasks.

When the **design** program finishes, the **interp** command prints out the best design found, translating from the internal to the original variables. The commands **iv**, **av** and **dv** respectively calculate the I-, A- and D-values of a design. The program automatically maintains a library of the best designs found.

Example. The following is an edited transcript of a session in which **gosset** was used to search for an I-optimal design with three continuous variables x, y, z, each between 0 and 1, for a full quadratic response surface, with 14 runs (a minimal design would use 10 runs), taking the best of 25 attempts.

```
10 range x y z 0 1
20 model (1+x+y+z)^2
compile
. . .
moments n=10^6
. . .
design type=I runs=14 n=25
. . .
interp
          x          y          z
        0.0000     0.0000     0.0000
        0.0000     0.5000     0.5000
        0.0000     1.0000     1.0000
        0.1707     0.0000     1.0000
        0.1707     1.0000     0.0000
        0.4742     0.5000     0.5000
        0.4742     0.5000     0.5000
        0.4742     0.5000     0.5000
        0.6630     0.0000     0.0000
        0.6630     1.0000     1.0000
        1.0000     0.0000     0.5712
        1.0000     0.4288     1.0000
        1.0000     0.5712     0.0000
        1.0000     1.0000     0.4288
iv
        0.4065171
```

Note that this design is symmetric in y and z but not x, and that a point close to the center of the cube appears three times in the design. This is one of many situations in which a probably optimal design could have been obtained from the built-in library.

We have printed the design rounded to four decimal places, but this is also under the user's control. This facility makes it possible to investigate the dependence of the I-value on the precision of the design points. One could for example print the design with only two decimal places of accuracy, and recalculate the I-value. (In this particular example the I-value increases only by about .000004.)

Further examples are given in the following section and in [51], [52], [53], [54]. The running time of the program, when searching for a minimal quadratic design in the k-dimensional ball, grows roughly as k^6.

We have recently extended the program by adding the ability to construct designs for situations in which the errors (in Eq. (1), for example), rather than being independent, have a known correlation matrix. We omit the details.

5. Examples

For over a year we have been running `gosset` to produce extensive tables of conjecturally optimal (or nearly optimal) designs for a variety of classical problems. Once found, these designs are incorporated into `gosset`'s built-in library. The families we have focused on are I-optimal designs for linear, quadratic or third-order models, in the ball or cube, with continuous or discrete (2- or 3-level) variables. Many D-optimal designs have also been computed.

At the present time the library includes third-order designs for the ball and cube with 4–20 runs (in 1 dimension), 10–30 runs (in 2 dimensions), 20–30 runs (in 3 dimensions), and 35–40 runs (in 4 dimensions), all using continuous variables, as well as third-order designs for the ball in some higher dimensions. Figure 1 shows the conjecturally I-optimal design with 12 runs in two dimensions.

These designs are very much smaller than the third-order designs available up to now (compare Bagchi [3], Huda [60], [61], and the references given there). For example the smallest four-dimensional design presented in these papers has 60 runs.

The parameters of the quadratic designs are shown in Table 1. The columns headed "ball" and "cube" refer to designs for the full quadratic model $\beta_0 + \sum \beta_i x_i + \sum\sum_{i \le j} \beta_{ij} x_i x_j$, in which the variables may range continuously over a k-dimensional ball or cube, while in the

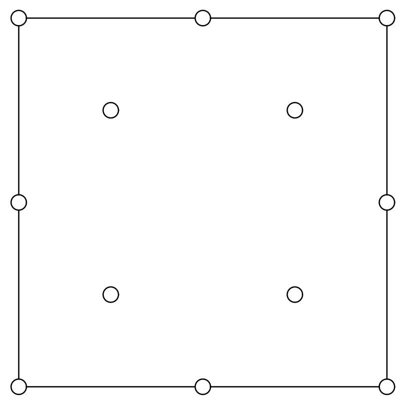

Figure 1: Conjecturally I-optimal 12-run design for cubic regression in square.

last column the variables are restricted to points with coordinates $-1, 0, +1$. For these three columns the modeling region is the continuous k-dimensional ball or cube. The column headed "Discrete (± 1)" refers to interaction designs, in which the variables are restricted to points with coordinates ± 1, the squared terms $\beta_{ii} x_i^2$ ($i = 1, \ldots, k$) are omitted from the model, and the modeling region is the same discrete set as the measurement region.

Table 1: Quadratic response surface designs in `gosset` library. (Entry gives number of runs.)

Variables k	Ball (Continuous)	Cube (Continuous)	Discrete (± 1)	Discrete ($0, \pm 1$)
1	3–30	3–30	2–6	3–7
2	6–21	6–40	4–8	6–10
3	10–33	10–30	7–11	10–14
4	15–28	15–22	11–15	15–19
5	21–33	21–29	16–20	21–25
6	28–39	28–36	22–26	28–32
7	36–54	36–44	29–33	36–40
8	45–70	45–53	37–41	45–49
9	55–62	55–63	46–50	55–59
10	66–74	66–74	56–60	66–70
11	78–86	78–86	67–71	78–82
12	91–99	91–98	79–83	91–95
13	105–113	105–113		
14	120–128	120–128		

The designs mentioned in the table are usually the best of at least 1000 attempts. Some of these designs (and the reasons for believing they are close to optimal) are briefly discussed in the following subsections.

We have made similar, although less extensive, tables of conjecturally D-optimal quadratic designs, some of which are mentioned in Section 5.2.

As to linear designs, at present the library contains conjecturally I- and D-optimal designs with $k+1$ runs for k variables taking values ± 1, for $k = 1, 2, \ldots, 20, 23, 27, 31, \ldots, 47$. For $k \equiv 3 \pmod 4$ these are Plackett-Burman-Rao designs, while the others are discussed in Section 5.3.

We have also used `gosset` to construct other classes of designs: mixture designs (cf. [22], [96]), block designs, and designs which are intended for use in circumstances when one of the measurements may be lost.

5.1. Quadratic designs in the ball

Study of the extensive collection of quadratic designs referred to in Table 1 has led us to make a number of observations about the properties of optimal designs in the ball.

Let $\mathcal{I}(k,n)$ (resp. $\mathcal{D}(k,n)$) denote any I-optimal (resp. D-optimal) design for a full quadratic model using n runs in a k-dimensional ball, for $k = 1, 2, \ldots$, and $n \geq (k+1)(k+2)/2$.

We begin with the observation that there appears to be a unique design $\mathcal{I}(3,14)$ (apart from orthogonal transformations), which consists of three copies of a point distant .003622 from the center of the ball and eleven points on the surface of the ball. The precise design is given in [53]. However, restricting $\mathcal{I}(3,14)$ to points at the center and on the surface of the ball incurs a loss of only about .00006% in I-efficiency. A similar phenomenon occurs for other values of k and n. In contrast, for D-optimal designs, it seems that the only points that occur are at the center and on the surface. We formalize these observations as follows.

Conjecture 1. *For all k and n, $\mathcal{D}(k,n)$ contains only points at the center and on the surface of the ball. This is not true for $\mathcal{I}(k,n)$, but restricting $\mathcal{I}(k,n)$ to designs with this property incurs a loss in I-efficiency of less than .01%.*

In the rest of this subsection we therefore restrict attention to designs supported only at the center and the surface of the ball. For any such design, not necessarily optimal, let B and C respectively denote the number of points on the surface (or boundary) and at the center, so that $n = B + C$. It can be shown (see the Appendix to [51]) that for the purpose of optimizing the placement of the B surface points one can take $C = 1$. The choice of number of center points and the arrangement of the surface points are independent problems.

Let ξ denote the discrete measure, normalized to have total measure 1, defined by the B surface points. If the moments of ξ up through order 4 agree with the moments of uniform measure on the surface of the sphere, the B surface points are said to form a *spherical 4-design* [26], [19, p. 89].

Theorem 1. *For fixed values of k, n, B and C, both the I- and D-efficiencies are maximized if the B surface points can be arranged to form a spherical 4-design.*

The proof for I-efficiency is given in [54], while for D-efficiency this follows from the work of Box and Hunter [14] and Kiefer [63] (see also [38], [80]).

If the surface points do form a spherical 4-design, the I- and D-values of the design can be calculated analytically, and we obtain the following bounds. The I-value of a design with n runs, B of which are on the surface and C at the center, is bounded below by

$$\frac{1}{(k+2)(k+4)}\left\{\frac{k^2(k^2+5k+10)}{2\beta}+\frac{8}{\gamma}\right\}, \tag{13}$$

and its D-value is bounded below by

$$\frac{k}{\beta}(k+2)^{(k-1)/(k+2)}\left\{\frac{\beta}{k\gamma 2^{k-1}}\right\}^{2/((k+1)(k+2))}, \tag{14}$$

where $\beta = B/n$, $\gamma = C/n = 1-\beta$.

One can now select the values of β and γ to minimize (13) or (14) for any fixed value of n. (Lucas [71] has studied how the efficiency of a 4-dimensional design changes as the number of center points is varied. `Gosset` automatically determines the best number of center points to use. The number of center points in I-optimal designs with $k \le 7$ can be found in Table 1 of [51].) For large n, (13) implies that in an I-optimal design the fraction of points at the center is

$$\gamma = \frac{4k\sqrt{k^2+5k+10}-16}{(k-1)(k+2)(k^2+4k+8)}. \tag{15}$$

(14) implies that in a D-optimal design this fraction is

$$\gamma = \frac{2}{(k+2)(k+4)}, \tag{16}$$

which is a theorem of Kiefer [63].

By substituting these expressions in (13) and (14) we obtain the limiting values

$$I_\infty = \frac{k+2}{2(k+4)}\left\{\frac{(k-1)(k^2+4k+8)}{k\sqrt{k^2+5k+10}-4}\right\}^2, \tag{17}$$

$$D_\infty = \frac{(k+1)(k+2)^{2k/(k+1)}}{k+3}\left(\frac{k+3}{2^k}\right)^{2/((k+1)(k+2))}. \tag{18}$$

Using (17) and (18) we can calculate the I- and D-efficiencies of any design via (10).

The expression (13) was discovered by fitting a formula to the I-values of the best designs. The explanation in terms of spherical 4-designs was found later. One of our reasons for believing that many of the designs found by the algorithm are optimal is that, in low dimensions $k \geq 2$, as n increases the I-values of the best designs fairly rapidly converge to the values given by (13). Table 2 for example shows the I-values of the best designs found in 3 and 4 dimensions, for designs using 1 center point and $n-1$ surface points. The columns headed Δ give the difference between I-value and the value given by (13) with $\gamma = 1/n$.

Table 2: Difference Δ between I-value of quadratic design in ball (with one center point) and lower bound (13).

Dimension 3			Dimension 4		
n	I-value	Δ	n	I-values	Δ
10	7.36919794	0.22634080	15	11.28634552	0.57205981
11	7.34766849	0.02481135	16	10.89510993	0.05066549
12	7.55627269	0.04458438	17	11.00693715	0.02777048
13	7.70714286	0.00000000	18	11.12656017	0.00891311
14	7.92232938	0.01463707	19	11.26928079	0.01002153
15	8.11224490	0.00000000	20	11.40360533	0.00009656
16	8.32382215	0.00382215	21	11.55000000	0.00000000
17	8.53035714	0.00000000	22	11.69841270	0.00000000
18	8.74285714	0.00000000	23	11.84848485	0.00000000
19	8.95714286	0.00000000	24	12.00000000	0.00000000
20	9.17293233	0.00000000	25	12.15277778	0.00000000
21	9.39000000	0.00000000	26	12.30666667	0.00000000

We see from Table 2 that in three dimensions the designs with $B = 12, 14$ and ≥ 16 surface points meet the bound, as do the four-dimensional designs with $B \geq 20$. In five dimensions the best designs with $B \geq 29$ surface points meet the bound; in six dimensions those with $B = 27, 36$ and ≥ 39; in seven dimensions those with $B \geq 53$; and in eight dimensions those with $B \geq 69$. These designs are usually not unique.

It follows from a result of Neumaier and Seidel [80] that a design has D-value given by (14) if and only if the B surface points form a spherical 4-design. (Incidentally, in reference to the discussion in Draper and Pukelsheim [33], this suggests that the difference between the D-value of a design and the minimum of (14) over all choices of $\gamma = C/n$ with $1 \leq C \leq n$ is a good measure of the departure of the design from rotatability.) There is a similar result for

I-values.

Theorem 2. *A quadratic design in the ball has I-value given by (13) if and only if the B surface points form a spherical 4-design.*

Thus we obtain spherical 4-designs from all the designs mentioned above. Furthermore, the nonexistence of designs meeting the bound (13) would imply the nonexistence of the corresponding spherical 4-designs. It therefore appears that spherical 4-designs with B points exist in three dimensions if and only if $B = 12, 14$ and ≥ 16; in four dimensions if and only if $B \geq 20$; in five dimensions if and only if $B \geq 29$; and so on. These conjectures are far more precise than our existing knowledge of spherical 4-designs (compare [26], [90], [4], [5], [84]). They are only conjectures, because in general we have only numerical evidence that the designs meet the bound (13). However, in several of these cases we have been able to prove that the points found by `gosset` are indeed spherical 4-designs (see [54]).

These spherical 4-designs yield rotatable designs when supplemented with the appropriate number of center points. For example, we have discovered an infinite family of 12-point three-dimensional spherical 4-designs (one of which is the icosahedron), which when supplemented by a center point form rotatable (and D-optimal) 13-point designs [54]. Similarly, any one of our 14-point designs, supplemented by two center points, forms a rotatable (and D-optimal) 16-point design. The design based on the icosahedron has of course a long history [14], [23], but the other designs appear to be new. Bose and Draper [8] give an infinite family of rotatable 16-point designs, but these lie on two concentric spheres, not one, and are neither D- nor I-optimal. The same is true of most of the rotatable designs constructed in [8], [30], [56].

Table 3 compares the I- and D-efficiencies of some designs found by our algorithm with those of earlier designs. In this table, a and b indicate Roquemore's designs 310 and 311B [86], c indicates the 12 vertices of the icosahedron, possibly supplemented with copies of the center point, and d is the central composite design with points $0, 0, 0$; $(\pm 1, 0, 0)$; $\pm 3^{-1/2}$, $\pm 3^{-1/2}$, $\pm 3^{-1/2}$ (the center and vertices of the rhombic dodecahedron). The designs mentioned in the first column consist of one of the sets of points shown in Table 4, supplemented by copies of the center point. To conserve space when specifying the points of a design, parentheses are used to indicate that all cyclic shifts of the enclosed coordinates are to be included. For example (abc) is an abbreviation for the three points abc, bca, cab. Square brackets have no special meaning and are used to group components; thus $\pm[ab]$ abbreviates the two points $+a$ $+b$ and $-a$ $-b$.

Table 3: I- and D-efficiencies of quadratic designs in 3-dimensional ball, comparing best designs found by `gosset` with earlier designs.

	`gosset` I-optimal				`gosset` D-optimal				Earlier designs			
n	C	B	I-eff.	D-eff.	C	B	I-eff.	D-eff.	C	B	I-eff.	D-eff.
10	1	9 (19)	89.6	97.9	1	9	89.6	97.9	1	9	80.7[a]	86.7[a]
11	2	9 (19)	96.4	95.4	1	10	89.8	99.7	1	10	89.0[b]	99.0[b]
12	2	10 (20)	99.3	98.0	1	11	87.3	99.4	–	–	–	–
13	2	11 (21)	98.5	98.4	1	12	85.6[c]	99.7[c]	1	12	85.6[c]	99.7[c]
14	3	11 (21)	98.8	95.1	1	13	83.3	99.3	2	12	98.5[c]	99.2[c]
15	3	12[c]	99.9	96.4	2	13	97.4	99.3	1	14	99.9[c]	99.1[d]

We see from Table 3 that indeed the conjecturally I-optimal designs do have the highest I-efficiency, and similarly for the conjecturally D-optimal designs (whose coordinates are not shown). It is worth remarking that for 13 points the icosahedron plus center is not I-optimal (although it appears to be D-optimal), while for 14 points the icosahedron plus two center points is neither I- nor D-optimal. The 15-point central composite design mentioned above is also neither I- nor D-optimal, nor is Doehlert's [28], [29], 13-point uniform shell design (not shown in the table), which consists of the center and vertices of a cuboctahedron, and has I- and D-efficiencies of only 82.6 and 97.0%.

Table 4: Conjecturally I-optimal quadratic designs in 3-dimensional ball (see Table 7).

Design (19) 9 surface points			Design (20) 10 surface points			Design (21) 11 surface points		
1.0000	0.0000	0.0000	±[0.0616	-0.5312]	0.8450	0.1079	0.4923	±0.8637
−0.5000	±0.8660	0.0000	±[0.8707	0.3746]	0.3186	0.4481	−0.3268	±0.8321
−0.7018	0.0000	±0.7123	±[0.7624	−0.6382]	0.1065	−0.5166	−0.6771	±0.5240
0.3509	±0.6080	±0.7123	±[0.1673	0.9191]	−0.3566	−0.8250	0.2847	±0.4882
			±[0.5779	−0.0525]	−0.8144	0.9517	0.3071	0.0000
						0.0000	1.0000	0.0000
						0.5255	−0.8508	0.0000

For larger numbers of runs in three dimensions, the best designs found by `gosset` appear to be rotatable and I-optimal for all $n \geq 20$, and rotatable and D-optimal for all $n \geq 18$.

In four dimensions we have computed designs with from 15 to 28 runs. Roquemore [86] gives three designs with 16 runs, the best of which (416C) has I-efficiency 90.3% and D-efficiency 96.9%. Our conjecturally D-optimal 16-point design has I-efficiency 92.3% and D-efficiency 99.7%, and our conjecturally I-optimal design has I-efficiency 94.3% and D-efficiency 99.1%. The latter design has a simple description by pairs of complex numbers. It consists of the points $[0,0]$, $[\omega^r, 0]$, $[0, \omega^r]$, $2^{-1/2}[-\omega^r, -\omega^s]$, where $\omega = e^{2\pi i/3}$, and $r, s \in \{0,1,2\}$.

For 27 points the program finds the Box-Behnken [10] design, which consists of three center points together with the vertices of the 24-cell [19], [23], and is I-optimal (although not unique).

Roquemore's six-dimensional design 628A, although it is not identified as such in [86], consists of a center point plus the well-known 27-point spherical 4-design associated with the Schläfli polytope (see [26], [20]). It is therefore both I- and D-optimal. This design was also found by `gosset` .

Box and Behnken [9], [10], Crosier [25], Doehlert [28], Doehlert and Klee [29] and others have given quadratic designs in higher-dimensional spheres, but these all contain many more points than ours.

5.2. Quadratic designs in the cube

In this section we discuss I- and D-optimal designs for quadratic models in the k-dimensional cube. Table 1 shows the parameters of the I-optimal designs we have considered.

We first discuss the two-dimensional problem. Box and Draper [15] used a hill-climbing method to search for D-optimal quadratic designs in the square with from 6 to 18 runs, and Haines [49] used simulated annealing to search for D-, I- and G-optimal designs in some low-dimensional problems. In particular, she found conjecturally optimal quadratic designs in the square with from 6 to 9 runs. Our program has confirmed the I- and D-optimality of the designs found in these two papers. (Although many of the designs presented in [49] and [15] are described as being respectively I- and D-optimal, we do not believe this optimality has been rigorously established.)

For three dimensions there are a number of papers dealing with D-optimal 3-level designs (see Atkinson [1], Mitchell and Bayne [75], [76]; Galil and Kiefer [41], [44]; Welch [97]; Meyer and Nachtsheim [72]). We have used `gosset` to search for both I- and D-optimal designs, both with continuous and with 3-level coordinates, and the designs with from 10 to 20 runs are summarized in Table 5. Four examples of the I-optimal continuous designs are shown in Table 6.

For the D-optimal 3-level designs our results agree with the above references. Comparing Tables 3 and 5 we see that in the cube there is a much greater difference between I- and D-optimal designs than in the ball. This is presumably because of the tendency for D-optimal designs to contain more points on the boundary and fewer at the center. We also see from

Table 5: Comparison of I- and D-efficiencies of best quadratic designs in 3-dimensional cube found by gosset (cts. means coordinates in range $[-1, 1]$; 3-lev. means coordinates are $0, \pm 1$).

	I-opt. cts.			I-opt. 3-lev.		D-opt. cts.		D-opt. 3-lev.	
n	Design	I-eff.	D-eff.	I-eff.	D-eff.	I-eff.	D-eff.	I-eff.	D-eff.
10	(22)	76.9	87.1	71.2	85.2	74.5	89.3	40.4	86.3
11	(23)	87.2	86.3	79.4	82.0	61.9	94.4	42.6	94.4
12	(24)	89.8	83.1	81.6	86.0	63.2	94.8	44.1	94.8
13		90.4	84.4	85.8	77.7	72.5	97.8	51.2	97.1
14	(25)	92.7	82.1	89.1	83.4	70.7	97.6	53.0	97.6
15		95.7	94.2	95.7	94.2	69.5	97.0	50.5	96.8
16		96.8	90.6	96.8	90.6	68.8	96.8	50.8	96.6
17		96.4	89.0	96.3	89.0	66.1	96.9	50.1	96.7
18		97.5	89.7	97.3	89.6	70.5	97.1	49.6	97.0
19		97.8	86.7	97.5	86.7	64.7	97.7	50.7	97.6
20		98.9	88.8	98.9	88.8	62.8	97.9	51.5	97.8

Table 5 that whereas for D-optimality it makes little difference whether we use 3-level or continuous coordinates, for I-optimality there is a considerable difference. With 12 runs for example we gain 8% in I-efficiency by using continuous coordinates. Furthermore, although the I-optimal designs have reasonably high D-efficiencies, the I-efficiencies of the D-optimal designs, especially the 3-level ones, are poor.

In dimensions 2 through 10, a number of authors have attempted to find D-optimal designs with the minimal number of runs in a cube. Draper and Lin [32] give a table comparing the D-values of designs found by Box and Draper [16], Draper and Lin [31], Lucas [69], Mitchell and Bayne [75], Notz [81] and Rechtschaffner [85]. We have reproduced this as Table 7, supplemented with an additional column showing the D-values of the designs found by gosset . In dimensions $k \geq 4$ the new designs are better. The entries give the reciprocals of the D-values, numbers which are to be made as large as possible.

For four dimensions, Meyer and Nachtsheim [72] reported that their simulated annealing algorithm found a 17-run design with $\det X'X = 1.4867 \cdot 10^{13}$ when the coordinates were restricted to three levels, but that they could only obtain $4.1296 \cdot 10^{12}$ when the points ranged over the whole cube. Gosset does not suffer from this weakness, and found an even better continuous design, with $\det X'X = 1.6863 \cdot 10^{13}$.

Table 6: Conjecturally I-optimal quadratic designs in 3-dimensional cube (see Table 5)

Design (22), 10 points				Design (23), 11 points			
1.0000	1.0000	1.0000		1.0000	1.0000	1.0000	
(0.9605	−0.1025	−0.1025)		(1.0000	−0.1905	−0.1905)	
(0.2553	−1.0000	−1.0000)		(0.2349	−1.0000	−1.0000)	
(−1.0000	1.0000	1.0000)		0.0589	0.0589	0.0589	
				(−1.0000	1.0000	1.0000)	
Design (24), 12 points				**Design (25), 14 points**			
1.0000	1.0000	1.0000		±[1.0000	1.0000]	−0.6585	
(1.0000	−0.3119	−0.3119)		±(1.0000	0.1423)	1.0000	
(0.1685	−1.0000	−1.0000)		(1.0000	−1.0000)	0.3261	
0.0925	0.0925	0.0925	twice	(1.0000	−1.0000)	−1.0000	
(−1.0000	1.0000	1.0000)		0.0000	0.0000	−0.0516	3 times
				0.0000	0.0000	−1.0000	

Table 7: Comparison of minimal quadratic designs in cube, based on Table 8 of Draper and Lin [32]. Entry gives reciprocal of D-value.

k	Draper and Lin [31]	Lucas [69]	Notz [81]	Mitchell and Bayne [75]	Box and Draper [16]	Rechtschaffner [85]	New designs
3	.303	.152	.400	.410	.423	.400	.423
4	.308	.096	.392	.425	.423	.392	.432
5	.241	.066	.459	.456	.374	.450	.467
6	.263	.048	.446		.317	.428	.464
7	.196	.036			.227	.383	.458
8	.321	.028			.193	.336	.455
9	.200	.023			.167	.293	.460
10	.165	.018			.146	.255	.465

5.3. Linear designs in the cube

There is a substantial body of work dealing with the construction of designs for linear regression in the k-dimensional cube (see for example [17], [34], [35], [36], [37], [39], [43]–[45], [46], [47], [73], [74], [77], [88], [89], [91], [98], and the references therein). The problem is to construct a design for the model $\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$, using $n \geq k+1$ points in the unit k-dimensional cube, either restricting the points to the vertices, or allowing them to range over the whole cube. For brevity we shall concentrate on the case of minimal (or saturated) designs, with $n = k+1$. Note that in the case where both the measurement and modeling regions consist of the set of vertices of the cube, the moment matrix M is the identity matrix, and A- and I-optimality coincide.

The problem of finding a minimal D-optimal design in which the points are restricted to the vertices of the cube can be rephrased as follows. Determine the value of $g(k+1)$, the maximal determinant of any $(k+1) \times (k+1)$ matrix with entries ± 1. Many optimal designs are known for this problem, and these provide an opportunity to evaluate the performance of our program in the case where all variables are discrete (which is the most difficult for the algorithm).

We find that although the algorithm performs well up to about 20 dimensions, finding the best designs known, in higher dimensions it does not perform as well as programs such as that of Smith [91] which are designed specifically for this problem.

Table 8 shows best lower bounds on $g(k+1)$ presently known. In this table, $*$ indicates that the entry is known to be the exact value of $g(k+1)$, H indicates a Hadamard matrix, and G that `gosset` was able to find this design.

Table 8: Best lower bound on maximal determinant of ± 1-matrix of order $k+1$; and normalized I-value of conjecturally I-optimal design.

k	$g(k+1)$	Notes	$I/(k+1)$
1	2	$*, G$	1
2	4	$*, G$	1.5
3	16	$*, H, G$	1
4	48	$*$, [34], [35], G	1.11111111
5	160	$*, G$	1.2
6	576	$*$, [98], G	1.27777778
7	4096	$*, H, G$	1
8	14336	$*$, [36], G	1.14795918
9	73728	$*$, [34], [35], G	1.11111111
10	327680	$*$, [36], [43], G	1.165
11	2985984	$*, H, G$	1
12	14929920	$*$, [34], [35], G	1.04
13	77635584	$*$, [34], [35], G	1.07692308
14	$\geq$418037760	[91], G	1.11142857
15	4294967296	$*, H, G$	1
16	21474836480	$*$, [77], G	1.06
17	146028888064	$*$, [34], [35], G	1.05882353
18	$\geq$894426939392	[91], G	1.09527929
19	10240000000000	$*, H, G$	1
20	59392000000000	$*$, [17], G	1.04447087
21	$\geq$377073157799936	[91]	
22	$\geq$2626567700217856	[91]	
23	36520347436056576	$*, H$	1

We also carried out a search for I-optimal (or equivalently A-optimal) designs with the same range of values of k. In each case we were able to find an I-optimal design which was also

D-optimal, and we conjecture that this will always be true for minimal designs. The converse is not always true. For $k = 10$ for example there are D-optimal designs which are not I-optimal (cf. [43, p. 1297]). The I-values of these designs, normalized by division by $k + 1$, are shown in the last column of Table 8.

For some small values of k the designs change dramatically when the points are not restricted to the vertices of the cube (and the measurement and modeling regions consist of the whole cube). For example, the following are conjecturally I-optimal designs for the cases $k = 2$ and 5:

$$k = 2: \quad \begin{bmatrix} + & + \\ a & - \\ - & a \end{bmatrix} \qquad k = 5: \quad \begin{bmatrix} + & + & + & + & + \\ b & - & - & + & + \\ - & b & + & + & - \\ - & + & b & - & + \\ + & + & - & b & - \\ + & - & + & - & b \end{bmatrix}$$

where $+ = +1$, $- = -1$, $a = 0.4391$, $b = -.2417$. Their I-values are respectively 2 and 3.1986, compared with 2.5 and 3.2 for the ± 1-designs mentioned in Table 8.

As a final example, we consider the problem of finding a D-optimal design for the case $k = 15$, $n = 19$, which Galil [39] mentions as unsolved. Equivalently, we wish to find a 19×16 ± 1-matrix X for which $\det X'X$ is maximal. The `gosset` program is

```
10 discrete x1 x2 ...  x15
20 model 1+x1+x2+...+x15
design runs=19
```

and the matrix shown in Table 9 is the best of 4000 tries. It has

$$\det X'X = 15447346721880801280 0 = 2^{54} 5^2 7^3 \ .$$

5.4. A "twisted" fractional factorial design

In the past year there have been numerous industrial and academic applications of `gosset` . There is space for only one example here. This problem arose in studying VLSI wafers, in an experiment with five quantitative discrete variables A, B, C, D, E taking the values -1 and 1. The experimenter had manufactured 14 wafers, in which A, B, C were set to $1, 1, 1$ (twice), $1, 1, -1$ (twice), and so on, excluding $-1, -1, -1$, but in which the values of D and E were not yet chosen. The response surface model involved main effects and all interactions except those

Table 9: 19×16 matrix X found by `gosset` , for which $\det X'X = 2^{54}5^27^3$.

1	−1	−1	−1	−1	−1	1	−1	1	1	−1	−1	1	1	1	−1
1	−1	−1	−1	−1	1	1	1	−1	−1	−1	−1	1	−1	1	1
1	−1	−1	1	−1	1	1	1	−1	−1	1	−1	−1	1	1	1
1	−1	−1	1	1	−1	−1	1	1	1	1	1	1	1	−1	1
1	−1	−1	1	1	1	−1	−1	1	−1	−1	1	−1	−1	1	−1
1	−1	1	−1	−1	1	1	1	1	−1	−1	1	1	−1	−1	1
1	−1	1	−1	1	−1	1	−1	−1	1	1	1	−1	−1	1	1
1	−1	1	−1	1	1	−1	−1	−1	−1	1	−1	1	1	−1	−1
1	−1	1	1	−1	−1	−1	1	−1	1	−1	−1	−1	−1	−1	−1
1	1	−1	−1	−1	−1	−1	−1	1	−1	1	−1	−1	−1	−1	1
1	1	−1	−1	−1	1	−1	1	−1	1	1	1	1	−1	1	−1
1	1	−1	−1	1	−1	1	1	−1	−1	−1	1	−1	1	−1	−1
1	1	−1	1	−1	1	1	1	−1	1	1	1	−1	1	−1	−1
1	1	−1	1	1	1	1	−1	−1	1	−1	−1	1	−1	−1	1
1	1	1	−1	1	1	−1	1	1	1	−1	−1	−1	1	1	1
1	1	1	1	−1	−1	−1	−1	−1	−1	−1	1	1	1	1	1
1	1	1	1	−1	1	1	−1	1	−1	1	−1	1	1	−1	−1
1	1	1	1	−1	1	1	−1	1	1	−1	1	−1	1	1	1
1	1	1	1	1	−1	1	1	1	−1	1	−1	1	−1	1	−1

mentioning C. The problem was to determine what settings for D and E should be used on these same 14 wafers.

We ran `gosset` with the program

```
10 discrete A B C D E -1 1
20 constraint A+B+C>-2.5
30 model 1+A+B+C+D+E+A*B+A*D+A*E+B*D+B*E+D*E
```

where line 20 is used to exclude the values $A = B = C = -1$, and asked for a design with 14 runs, guessing that this would produce a design meeting the requirements. It did. In fact we obtained the same design whether we specified I-, D- or A-optimality. Examination of the computer output showed that this presumably I-, D- and A-optimal design can be regarded as a "twisted" fractional factorial design with the following definition: take all vectors A, B, C, D, E which satisfy $ABCDE = -1$, except that if $A = B = -1$, $C = 1$ the rule is $ABCDE = 1$, and then discard the two vectors with $A = B = C = -1$. Note that in the statement of the problem C is treated differently from A and B, and the solution shows the same asymmetry. This experiment has now been completed.

We are grateful to our colleagues L. Denby, A. E. Freeny and J. M. Landwehr for telling

us about this problem.

5.5. Designs for extrapolation

There have been a number of papers on designs for extrapolation, but these deal mostly with the asymptotic theory ([42], [57], [58], [66], [67], [68]). In this section we give some examples of minimal or close to minimal designs.

These designs have not yet found application, but are included to illustrate the use of `gosset` in situations in which the regions O and R are quite distinct, even disjoint. Such designs must be used with care, for now it is harder to justify a polynomial model. However, they illustrate one of the program's more unusual features, the technique could be modified to accommodate other models (it would be easy to modify `gosset` to allow rational or trigonometric models), and the first of these designs could arise in several situations.

The first example asks for a design which will fit a quadratic response surface to the radiation level in a room with coordinates $-1 \leq x \leq 1$, $0 \leq y \leq 1$, $0 \leq z \leq 1$, with the constraint that because of the high radiation level in the left half of the room, measurements can only be made in the right half, i.e. in the region $0 \leq x, y, z \leq 1$. The `gosset` program is

```
10 range x y z 0 1
20 range x′ -1 1
30 range y′ z′ 0 1
40 model (1+x′+y′+z′)^2
```

An example of a conjecturally I-optimal design with 13 runs is shown in Table 12.

Table 10: A 13-run design for fitting a quadratic model in a room from measurements made only in one half.

x	y	z
0.0000	0.0000	0.0000
0.0000	(0.0323	1.0000)
0.0000	0.5002	0.5002
0.0000	0.8162	0.8162
0.5124	1.0000	1.0000
0.5132	(0.0000	0.4361)
0.5187	(0.0000	0.8448)
1.0000	0.0000	0.0000
1.0000	(0.3695	0.9636)

In another setting for this example, the x-coordinate is time, and we wish to make measurements over a 24-hour period in order to model a response over 48 hours.

Many variations of these designs are possible. An extreme example might arise in astronomy — find an I-optimal design for modeling a quadratic response on the moon from measurements made on or in the earth. The `gosset` program is

```
10 sphere x y z radius 4000
20 sphere x′ y′ z′ radius 1000 center 240000 0 0
30 model (1+x′+y′+z′)^2
```

and a 10-run (minimal) design consists of the points $[3997, -154.0, 37.5]$, $[3992, 110.0, 230.7]$, $[3991, 42.0, -268.1]$, $[159, 158.9, -28.0]$, $[154, -202.4, 47.7]$, $[151, -107.2, 36.1]$, $[136, 135.9, 200.3]$, $[133, 16.4, -259.4]$, $[-3997, 164.0, -7.0]$, $[-3997, -164.9, 5.8]$.

Finally, we consider a 1-dimensional extrapolation problem used as an illustration by Hoel and Levine [58]: construct a design on $[-1, 1]$ to estimate a cubic polynomial at the single point 2. It is shown in [58] that the I-optimal design places respectively 5/52, 12/52, 20/52 and 15/52 of the observations at the points -1, $-1/2$, $1/2$ and 1. We used the `gosset` program

```
10 range x
20 range x′ 1.99999 2.00001
30 model (1+x)^3
```

to produce comparable designs, some of which are shown in Table 13.

Table 11: Designs for extrapolation of cubic polynomial at a single point. (n is number of observations.)

n	I-eff.	Design			
4	87.51	-1,	$-.5798$,	$.4689$,	1
5	91.81	-1,	$-.5367$,	$.4764^2$,	1
6	94.07	-1,	$-.5291$,	$.5213^2$,	1^2
10	99.50	-1,	$-.4979^2$,	$.5119^4$,	1^3
20	99.32	-1^2,	$-.5025^5$,	$.4854^8$,	1^5
52	99.29	-1^5,	$-.4994^{13}$,	$.4861^{21}$,	1^{13}

We see that efficient designs are obtained even with small numbers of observations. Incidentally, although in the asymptotic theory Hoel and Levine show [58] that the same observation points can be used for extrapolation to any single point, our program shows that in the finite theory this is not true.

The last two lines of the table show one of our algorithm's limitations: because of the way we move the design points, they all attempt to move to the most favorable position, which tends to make them fall into "bunches". The points in a bunch then stay together throughout the iteration process. If there are a large number of points, many more than in a minimal design, it may happen (as we have seen here) that there is a vanishingly small chance of stumbling on the optimal distribution into bunches. This does not occur with smaller numbers of observations, because then every distribution has a significant probability of being tried.

Acknowledgments

We are extremely grateful to David H. Doehlert, whose letter to us (see Sect. 1) triggered this work, and who has since provided invaluable guidance and encouragement. During the course of this work we have also benefited from discussions with many of our colleagues at Bell Labs. In particular we should like to thank A. R. Calderbank, A. E. Freeny, C. L. Mallows, V. N. Nair and D. Pregibon for their advice. We are also grateful for comments and suggestions from R. B. Crosier, J. M. Lucas, A. B. Owen and R. D. Tobias.

References

[1] Atkinson, A. C. (1973), "Multifactor Second Order Designs for Cuboidal Regions," *Biometrika*, *60*, 15–19.

[2] Atwood, C. L. (1969), "Optimal and Efficient Designs of Experiments," *Annals of Mathematical Statistics, 40*, 1570–1602.

[3] Bagchi, S. (1986), "A Series of Nearly D-Optimal Third Order Rotatable Designs," *Sankhyā*, B, *48*, 186–198.

[4] Bajnok, B. (1991), "Construction of Spherical 4- and 5-Designs," *Graphs and Combinatorics, 7*, 219–233.

[5] Bajnok, B. (1992), "Construction of Spherical t-Designs," *Geometriae Dedicata*, in press.

[6] Becker, R. A., Chambers, J. M., and Wilks, A. R. (1988), *The New S Language*, Wadsworth and Brooks, Pacific Grove, California.

[7] Beightler, C. S., Phillips, D. T., and Wilde, D. J. (1979), *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, New York, 2nd ed.

[8] Bose, R. C. and Draper, N. R. (1959), "Second Order Rotatable Designs in Three Dimensions," *Annals Math. Stat., 30*, 1097–1112.

[9] Box, G. E. P. and Behnken, D. W. (1960a), "Simplex-Sum Designs: a Class of Second Order Rotatable Designs Derivable from those of First Order," *Ann. Math. Stat., 31*, 838–864.

[10] Box, G. E. P. and Behnken, D. W. (1960b), "Some New Three Level Designs for the Study of Quantitative Variables," *Technometrics*, *2*, 455–475.

[11] Box, G. E. P. and Draper, N. R. (1959), "A Basis for the Selection of a Response Surface Design," *J. American Statistical Association, 54*, 622–654.

[12] Box, G. E. P. and Draper, N. R. (1963), "The Choice of a Second Order Rotatable Design," *Biometrika 50,*, 335–352.

[13] Box, G. E. P. and Draper, N. R. (1987), *Empirical Model-Building and Response Surfaces*, Wiley, New York.

[14] Box, G. E. P. and Hunter, J. S. (1957), "Multi-Factor Experimental Designs for Exploring Response Surfaces," *Annals of Mathematical Statistics, 28*, 195–241.

[15] Box, M. J. and Draper, N. R. (1971), "Factorial Designs, the $|X'X|$ Criterion, and Some Related Matters," *Technometrics 13*, 731–742.

[16] Box, M. J. and Draper, N. R. (1974), "On Minimum-Point Second-Order Designs," *Technometrics, 16*, 613–616.

[17] Chadjipantelis, T., Kounias, S., and Myssiadis, C. (1987), "The Maximum Determinant of 21×21 $(+1, -1)$-Matrices and D-Optimal Designs," *J. Stat. Planning Inference, 16*, 167–178.

[18] Chambers, J. M. and Hastie, T. J., editors (1991), *Statistical Models in S*, Wadsworth and Brooks, Pacific Grove, California.

[19] Conway, J. H. and Sloane, N. J. A. (1988), *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York.

[20] Conway, J. H. and Sloane, N. J. A. (1991), "The Cell Structures of Lattices," in *Miscellanea mathematica*, P. Hilton et al., editors, Springer-Verlag, New York, pp. 71–107.

[21] Cook, R. D. and Nachtsheim, C. J. (1980), "A Comparison of Algorithms for Constructing Exact D-Optimal Designs," *Technometrics, 22*, 315–324.

[22] Cornell, J. A. (1990), *Experiments with Mixtures*, Wiley, New York, 2nd edition.

[23] Coxeter, H. S. M. (1973), *Regular Polytopes*, Dover, New York, 3rd ed.

[24] Crary, S. B. (1991), "Optimal Design of Experiments for Sensor Calibration," in *Proc. 1991 Internat. Conf. Solid-State Sensors and Actuators (San Francisco, June 23–27, 1991)*.

[25] Crosier, R. B. (1991), *Some New Three-Level Response Surface Designs*, Report CRDEC-TR-308, U.S. Army Chemical Research, Development & Engineering Center, Aberdeen Proving Ground, Maryland.

[26] Delsarte, P., Goethals, J. M., and Seidel, J. J. (1977), "Spherical Codes and and Designs," *Geometriae Dedicata, 6*, 363–388.

[27] Dodge, Y., Fedorov, V. V. and Wynn, H. P. (1988), "Optimal Design of Experiments: An Overview," in *Optimal Design and Analysis of Experiments*, ed. Y. Dodge et al., North-Holland, Amsterdam, pp. 1–11.

[28] Doehlert, D. H. (1970), "Uniform Shell Designs," *J. Roy. Statist. Soc.*, Ser. C, *19*, 231–239.

[29] Doehlert, D. H. and Klee, V. L. (1972), "Experimental Designs Through Level Reduction of the d-Dimensional Cuboctahedron," *Discrete Math., 2*, 309–334.

[30] Draper, N. R. (1960), "Second Order Rotatable Designs in Four or More Dimensions," *Annals Math. Stat., 31*, 23–33.

[31] Draper, N. R. and Lin, D. K. J. (1988), *Using Plackett and Burman Designs With Fewer Than $N-1$ Factors*, Technical Report 848, Department of Statistics, Univ. of Wisconsin, Madison, Wisconsin.

[32] Draper, N. R. and Lin, D. K. J. (1990), "Small Response-Surface Designs," *Technometrics, 32*, 187–194.

[33] Draper, N. R. and Pukelsheim, F. (1990), "Another Look at Rotatability," *Technometrics, 32*, 195–202.

[34] Ehlich, H. (1964a), "Determinantenabschätzungen für Binäre Matrizen," *Math. Zeit, 83*, 123–132.

[35] Ehlich, H. (1964b), "Determinantenabschätzungen für Binäre Matrizen mit $n \equiv 3 \bmod 4$," *Math. Zeit, 84*, 438–447.

[36] Ehlich, H. and Zeller, K. (1962), "Binäre Matrizen," *Zeit. Angew. Math. Mech., 42*, 20–21.

[37] Farmakis, N. (1991), "Constructions of A-Optimal Weighing Designs When $n = 19$," *J. Stat. Planning Inference*, *27*, 249–261.

[38] Farrell, R. H., Kiefer, J. and Walbran, A. (1967), "Optimum Multivariate Designs," in *Proc. 5th Berkeley Sympos. Math. Statist. and Probability*, Univ. Calif. Press, Berkeley, Calif., *1*, pp. 113–138.

[39] Galil, Z. (1985), "Computing D-Optimum Weighing Designs: Where Statistics, Combinatorics, and Computation Meet," in *Proceedings Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer*, L. M. Le Cam and R. A. Olshen, eds., Wadsworth, Monterey, Calif., Vol. II, pp. 635–650.

[40] Galil, Z., and Kiefer, J. (1977a), "Comparison of Rotatable Designs for Regression on Balls, I (Quadratic)," *J. Statistical Planning and Inference, 1*, 27–40.

[41] Galil, Z., and Kiefer, J. (1977b), "Comparison of Design for Quadratic Regression on Cubes," *J. Statistical Planning and Inference, 1*, 121–132.

[42] Galil, Z. and Kiefer, J. (1979), "Extrapolation Designs and Φ_p-Optimum Designs for Cubic Regression on the q-Ball," *J. Stat. Planning Inference, 3*, 27–38.

[43] Galil, Z. and Kiefer, J. (1980a), "D-Optimum Weighing Designs," *Annals of Statistics, 8*, 1293–1306.

[44] Galil, Z. and Kiefer, J. (1980b), "Time- and Space-Saving Computer Methods, Related to Mitchell's DETMAX, for Finding D-Optimum Designs," *Technometrics, 22*, 301–313.

[45] Galil, Z. and Kiefer, J. (1980c), "Optimum Weighing Designs," in *Recent Developments in Statistical Inference and Data Analysis*, K. Matusita, editor, North-Holland, Amsterdam, pp. 183–189.

[46] Galil, Z. and Kiefer, J. (1982a), "On the Characterization of D-Optimum Weighing Designs for $n \equiv 3(\mathrm{mod}4)$," in *Statistical Decision Theory and Related Topics III*, S. S. Gupta and J. O. Berger, eds., Academic Press, New York, Vol. 1, pp. 1–35.

[47] Galil, Z. and Kiefer, J. (1982b), "Construction Methods for D-Optimum Weighing Designs When $n \equiv 3(\mathrm{mod}4)$," *Annals of Statistics, 10*, 502–510.

[48] Giovannitti-Jensen, A. and Myers, R. H. (1989), "Graphical Assessment of the Prediction Capability of Response Surface Designs," *Technometrics, 31*, 159–171.

[49] Haines, L. M. (1987), "The Application of the Annealing Algorithm to Construction of Exact Optimal Designs for Linear Regression Models," *Technometrics, 29*, 439–447.

[50] Hardin, R. H. and Sloane, N. J. A. (1992), *Operating Manual for Gosset: A General-Purpose Program for Constructing Experimental Designs* (*Second Edition*), Statistics Research Report, AT&T Bell Laboratories, Murray Hill, New Jersey.

[51] Hardin, R. H. and Sloane, N. J. A. (1992a), *Computer-Generated Minimal (and Larger) Response-Surface Designs: (I) The Sphere*, Statistics Research Report, AT&T Bell Laboratories, Murray Hill, New Jersey.

[52] Hardin, R. H. and Sloane, N. J. A. (1992b), *Computer-Generated Minimal (and Larger) Response-Surface Designs: (II) The Cube*, Statistics Research Report, AT&T Bell Laboratories, Murray Hill, New Jersey.

[53] Hardin, R. H. and Sloane, N. J. A. (1992c), *Tables of Experimental Designs*, Statistics Research Report, AT&T Bell Laboratories, Murray Hill, New Jersey.

[54] Hardin, R. H. and Sloane, N. J. A. (1992d), "New Spherical 4-Designs," *Discrete Math.*, to appear.

[55] Hardin, R. H., Sloane, N. J. A., and Smith, Warren D. (1992), *Spherical Codes*, in preparation.

[56] Herzberg, A. M. (1967), "A Method for the Construction of Second Order Rotatable Designs in k Dimensions," *Annals Math. Stat., 38*, 177–180.

[57] Hoel, P. G. (1965), "Optimum Designs for Polynomial Extrapolation," *Annals Math. Stat., 36*, 1483–1493.

[58] Hoel, P. G. and Levine, A. (1964), "Optimal Spacing and Weighting in Polynomial Prediction," *Annals Math. Stat., 35*, 1553–1560.

[59] Hooke, R. and Jeeves, T. A. (1961), "'Direct Search' Solution of Numerical and Statistical Problems," *Journal Association for Computing Machinery, 8*, 212–229.

[60] Huda, S. (1982), "Some Third-Order Rotatable Designs," *Biom. J., 24*, 257–263.

[61] Huda, S. (1983), "Two Third-Order Rotatable Designs in Four Dimensions," *J. Stat. Planning Inference, 8*, 241–243.

[62] Hume, A. G. and McIlroy, M. D. (1990), *UNIX Research System, Tenth edition*, AT&T Bell Laboratories, Murray Hill, New Jersey, Vol. II.

[63] Kiefer, J. (1960), "Optimum Experimental Designs V, With Applications to Systematic and Rotatable Designs," in *Proc. 4th Berkeley Sympos. Math. Statist. and Probability*, Univ. Calif. Press, Calif., *1*, pp. 381–405.

[64] Kiefer, J. C. (1985), *Collected Papers III: Design of Experiments*, L. D. Brown et al., editors, Springer-Verlag, New York.

[65] Kiefer, J. and Wolfowitz, J. (1960), "The Equivalence of Two Extremum Problems," *Canadian Journal of Mathematics, 12*, 363–366.

[66] Kiefer, J. and Wolfowitz, J. (1964a), "Optimum Extrapolation and Interpolation Designs I," *Annals Inst. Statist. Math.*, *16*, 79–108.

[67] Kiefer, J. and Wolfowitz, J. (1964b), "Optimum Extrapolation and Interpolation Designs II," *Annals Inst. Statist. Math.*, *16*, 295–303.

[68] Kiefer, J. and Wolfowitz, J. (1965), "On a Theorem of Hoel and Levine on Extrapolation Designs," *Annals Math. Stat., 36*, 1627–1655.

[69] Lucas, J. M. (1974), "Optimum Composite Designs," *Technometrics, 16*, 561–567.

[70] Lucas, J. M. (1976), "Which Response Surface Design is Best," *Technometrics, 18*, 411–417.

[71] Lucas, J. M. (1977), "Design Efficiencies for Varying Numbers of Centre Points," *Biometrika, 64*, 145–147.

[72] Meyer, R. H. and Nachtsheim, C. J. (1988), "Constructing Exact D-Optimal Experimental Designs by Simulated Annealing," *Amer. J. Mathematical and Management Sciences, 8*, 329–359.

[73] Mitchell, T. J. (1974a), "An Algorithm for the Construction of 'D-Optimal' Experimental Designs," *Technometrics, 16*, 203–210.

[74] Mitchell, T. J. (1974b), "Computer Construction of 'D-Optimal' First-Order Designs," *Technometrics, 16*, 211–220.

[75] Mitchell, T. J. and Bayne, C. K. (1976), *D-Optimal Fractions of Three Level Factorial Designs*, Report CSD-19, Oak Ridge National Laboratory, Oak Ridge, Tenn.

[76] Mitchell, T. J. and Bayne, C. K. (1978), "D-Optimal Fractions of Three-Level Factorial Designs," *Technometrics, 20*, 369–380.

[77] Moyssiadis, C. and Kounias, S. (1982), "The Exact D-Optimal First Order Saturated Design with 17 Observations," *J. Stat. Plannings Inference, 7*, 13–27.

[78] Myers, R. H. Khuri, A. I., and Carter, W. H., Jr. (1989), "Response Surface Methodology: 1966–1988," *Technometrics, 31*, 137–157.

[79] Nachtsheim, C. I. (1987), "Tools for Computer-Aided Design of Experiments," *J. Qual. Technol., 19*, 132–160.

[80] Neumaier, A. and Seidel, J. J. (1992), "Measures of Strength $2e$, and Optimal Designs of Degree e," *Sankhyā*, to appear.

[81] Notz, W. (1982), "Minimal Point Second Order Designs," *Journal Statistical Planning and Inference, 6*, 47–58.

[82] Pearson, E. S. and Wishart, J., editors (1942), *"Students" Collected Papers*, University College London.

[83] Pesotchinsky, L. L. (1975), "D-Optimum and Quasi-D-Optimum Second-Order Designs on a Cube," *Biometrika, 62*, 335–340.

[84] Rabau, P. and Bajnok, B. (1991), "Bounds for the Number of Nodes in Chebyshev Type Quadrative Formulas," *J. Approx. Theory, 67*, 199–214.

[85] Rechtschaffner, R. L. (1967), "Saturated Fractions of 2^n and 3^n Factorial Designs," *Technometrics, 9*, 569–575.

[86] Roquemore, K. G. (1976), "Hybrid Designs for Quadratic Response Surfaces," *Technometrics, 18*, 419–423.

[87] St. John, R. C. and Draper, N. R. (1975), "D-Optimality for Regression Designs: A Review," *Technometrics, 17*, 15–23.

[88] Sathe, Y. S. and Shenoy, R. G. (1989), "A-Optimal Weighing Designs When $N \equiv 3(\bmod 4)$", *Annals Statistics*, *17*, 1906–1915.

[89] Sathe, Y. S. and Shenoy, R. G. (1991), "Further Results on Construction Methods for Some A- and D-Optimal Weighing Designs When $N \equiv 3(\bmod 4)$", *J. Stat. Planning Inference, 28*, 339–352.

[90] Seymour, P. D. and Zaslavsky, T. (1984), "Averaging Sets: A Generalization of Mean Values and Spherical Designs," *Advances in Math., 52*, 213–240.

[91] Smith, Warren D. (1988), *Studies in Computational Geometry Motivated by Mesh Generation*, Ph.D. Dissertation, Princeton Univ., 1988.

[92] Snow, M. E. and Crary, S. B. (1991), "The Use of Simulated Annealing in the I-Optimal Design of Experiments," *The Michigan Academician*, submitted.

[93] Steinberg, D. M. and Hunter, W. G. (1984), "Experimental Design: Review and Comment," *Technometrics, 26*, 71–97.

[94] Studden, W. J. (1977), "Optimal Designs for Integrated Variance in Polynomial Regression," in *Statistical Decision Theory and Related Topics II*, ed. S. S. Gupta and D. S. Moore, Academic Press, pp. 411–420.

[95] Vining, G. G. (1990), *A Computer Program for Generating Variance Dispersion Graphs*, Technical Report 373, Dept. Statistics, Univ. Florida, Gainesville, FL.

[96] Vuchkov, I. N., Damgaliev, D. L. and Yontchev, C. A., (1981), "Sequentially Generated Second Order Quasi D-Optimal Designs for Experiments with Mixture and Process Variables," *Technometrics, 23*, 233–238. Springer-Verlag, New York.

[97] Welch, W. J. (1982), "Branch-and-Bound Search for Experimental Designs Based on D-Optimality and Other Criteria," *Technometrics, 24*, 41–48.

[98] Williamson, J. (1946), "Determinants Whose Elements are 0 and 1," *Amer. Math. Monthly, 53*, 427–434.

[99] Yonchev, H. (1988), "New Computer Procedures for Constructing D-Optimal Designs," An Overview," in *Optimal Design and Analysis of Experiments*, ed. Y. Dodge et al., North-Holland, Amsterdam, pp. 71–80.