REFERENCES

TECHNOMETRICS, FEBRUARY 1995, VOL. 37, NO. 1

The Coordinate-Exchange Algorithm for Constructing Exact Optimal Experimental Designs

Ruth K. MEYER

Department of Business Computer
Information Systems
St. Cloud State University
St. Cloud, MN 56301

Christopher J. NACHTSHEIM

Operations and Management
Science Department
Curtis L. Carlson School of
Management
University of Minnesota
Minneapolis, MN 55455

We describe the cyclic coordinate-exchange algorithm for constructing D-optimal and linear-optimal experimental designs. The algorithm uses a variant of the Gauss–Southwell cyclic coordinate-descent algorithm within the k-exchange algorithm to achieve substantive reductions in required computing. Among its advantages are the following: Candidate sets, which grow exponentially in the number of factors, need not be explicitly constructed or enumerated. Convex design spaces (or mixed convex by discrete design spaces) are handled directly, without the need for sophisticated nonlinear programming routines or candidate-set adjustment. For design problems having 10 or more factors, the reductions in execution time are typically two or more orders of magnitude when compared to standard candidate-set-based procedures such as k exchange, yet the designs produced exhibit no loss of efficiency.

The use of optimal experimental designs has undergone substantial growth in recent years due to the increasing availability of software. Echip, SAS, RS/Discover, and Design Expert are examples of systems that provide optimal-design components. See Nachtsheim (1987) and Koch, Morris, Nachtsheim, and Welch (1991) for related discussions. Unfortunately, the size of problems that can be solved by such systems is limited because the computational intensity required grows rapidly with the number of factors (Meyer 1989). Even on most mainframe computers, attempting to construct optimal experiments for problems having more than 10 factors is typically a futile experience. In this article, we introduce the *cyclic coordinate-exchange* algorithm for constructing optimal experimental designs that is particularly effective in large problems. Among its benefits are the following:

1. For factorial design spaces, candidate sets (which grow exponentially in the number of factors) need not be explicitly constructed or enumerated. For example, the normal candidate set for a q-factor, three-level experiment consists of 3^q candidate points.
2. For convex design spaces, problems stemming from the need to identify global optima of the variance function (see Cook and Nachtsheim 1980; Johnson and Nachtsheim 1983) are completely avoided.
3. Mixed quantitative × qualitative (i.e., discrete and continuous) design spaces are handled trivially.
4. A reduction in computing time of one or two orders of magnitude is realized for large problems, giving comparable results in terms of the efficiencies of the designs constructed.

Although the exchange procedure described is easily adapted to most criteria, we describe implementations for the determinant (D) and linear criteria (L). Well-known special cases of linear criteria are trace-optimality (A), weighted trace, and integrated variance (I).

After introducing notation and briefly reviewing the literature in Section 1, we give details of the algorithm in Section 2. Its performance is evaluated for linear and quadratic models in Section 3, and two quality-improvement experiments are considered in Section 4. Concluding comments are given in Section 5.

A general linear model of the form $y = X\beta + \varepsilon$ is assumed, where y is an $n \times 1$ vector of observations, X is the $n \times p$ design matrix, β is a $p \times 1$ vector of unknown regression parameters, and ε is an $n \times 1$ vector of uncorrelated random errors with mean 0 and, with no loss of generality, variance $\sigma^2 = 1$. We assume throughout that X is of full rank and use D to denote the variance–covariance matrix, $(X'X)^{-1}$, of the ordinary least squares estimate of β. The ith observation, $y_i = f'(x_i)\beta + \varepsilon_i, i = 1, 2, \ldots, n$, is obtained at x_i, a point in a q-dimensional closed compact design space χ, where $f'(x_i)$ is the corresponding row of X. The covariance between fitted values at x_i and x_j is proportional to the *covariance function*,

$v(x_i, x_j) = f'(x_i)Df(x_j)$, which yields the *variance* function $v(x_i)$ for $i = j$.

D-optimal designs maximize the determinant of the information matrix, $|X'X|$. A design is linear-optimal if it minimizes some linear functional L on $(X'X)^{-1}$. L must be linear in the sense that $L(A + B) = L(A) + L(B)$ and $L(cA) = cL(A)$ for matrices A and B and scalar c. L must also be nonnegative in the sense that $L(A) \geq 0$ for positive semidefinite A. As indicated, some popular linear criteria are A or trace optimality: $L((X'X)^{-1}) = \text{tr}(D)$ and the integrated variance criterion

$$L((X'X)^{-1}) = \int_{\chi} v(x)dx = \text{tr}(DM),$$

where

$$M = \int_{\chi} f(x)f'(x)dx.$$

1. EXACT DESIGN ALGORITHMS

A variety of approaches have been used to construct optimal designs. For example, Box and Draper (1971) used Powell's (1964) derivative-free direct search to construct D-optimal factorial designs. Welch's (1982) branch and bound guarantees a catalog of D-optimal designs on a finite design space. Haines (1987) and Meyer and Nachtsheim (1988) used simulated annealing to construct D-optimal designs for continuous design spaces. Currently, all commercially available software systems use *exchange algorithms*. Exchange algorithms begin with n-point starting designs. Typically then, new points from χ are added and existing design points are deleted in an effort to improve the value of the objective function. Rank-1 augmentation algorithms choose points to add and delete sequentially. Examples are Wynn's algorithm (Wynn 1972) and DETMAX (Mitchell 1974). Rank-2 exchange algorithms choose points to add and delete simultaneously. Examples are Fedorov's algorithm (Fedorov 1972), the modified Fedorov algorithm (Cook and Nachtsheim 1980), the k-exchange algorithm (Johnson and Nachtsheim 1983), and the kl-exchange algorithm (Atkinson and Donev 1989). Although the works just cited deal almost exclusively with D optimality, some linear optimal implementations have been described. See, for example, Welch (1984) and Fedorov (1972).

Because the cyclic coordinate-exchange procedure, as described in Section 2, is a modification of the k-exchange algorithm, a more detailed description of the latter algorithm is needed. Fedorov (1972) showed that if the ith point in the current design is exchanged for an arbitrary $x \in \chi$, the multiplicative change to $|X'X|$ is given by the *delta function*

$$\Delta_D(x_i, x) = 1 + [v(x) - v(x_i)] + [v^2(x, x_i) - v(x)v(x_i)], \quad (1)$$

where the subscript D is used to emphasize the tie to D optimality. This is the basis of Fedorov's algorithm: At stage k, find the points x_i^* and x^*, which maximize $\Delta_D(x_i, x)$, and make the indicated exchange. The algorithm can be computationally expensive because up to n optimizations over χ—one for each distinct x_i—are required at each stage. The k-exchange algorithm reduces the number of optimizations required by focusing on the k least critical points in the design. For example, if x_i is deleted, the multiplicative increase in $|D|$ is $[1 - v(x_i)]^{-1}$. Thus we consider for exchange those design points having the k smallest values of the *deletion function*,

$$d_D(x_i) = v(x_i). \quad (2)$$

The user-specified integer k dictates the intensity of the search. If $k = 1$, a form of Wynn's algorithm (Wynn 1972) emerges. For $k = n$, the modified Fedorov algorithm (Cook and Nachtsheim 1980) results. Empirical experience suggests that $k \leq n/4$ (Cook and Nachtsheim 1982) is often sufficient to give designs comparable to the modified Fedorov algorithm.

A k-exchange algorithm for linear optimality is easily formulated. The *phi function* is given by

$$\phi(x_i, x_j) = L(Df(x_i)f'(x_j)D), \quad (3)$$

and we use $\phi(x, x) = \phi(x)$ for brevity. Suppose that the ith point in the current design, x_i, is exchanged for $x \in \chi$. The decrease to the objective function, $L[D]$, resulting from such an exchange is given by

$$\Delta_L(x_i, x) = \frac{[1 - v(x_i)]\phi(x) + v(x, x_i)[\phi(x, x_i) + \phi(x_i, x)] - [1 - v(x)]\phi(x_i)}{\Delta_D(x_i, x)} \quad (4)$$

(Fedorov 1972). The k points for deletion are those having the k smallest values of the deletion function

$$d_L(x_i) = \frac{\phi(x_i)}{1 - v(x_i)}. \quad (5)$$

Details are provided in Appendix A.

Both versions of the k-exchange algorithm require that D be computed at the start and updated following an exchange. Standard rank-2 updating formulas are applicable (Fedorov 1972). Using "up" to denote the updated form of D after an exchange has taken place, we have

$$D_{\text{up}} = D - DF_1[I_2 + F_2'DF_1]^{-1}F_2'D, \quad (6)$$

where $F_1 = [f(x), -f(x_i)]$ and $F_2 = [f(x), f(x_i)]$. Updating D thus requires taking the inverse of a 2×2 matrix and some matrix multiplication. This avoids having to recompute $X'X$ and its inverse from scratch with each exchange.

2. CYCLIC COORDINATE EXCHANGE ALGORITHM

2.1 Exchanging Coordinate Subsets

Exact optimal designs solve the following nonlinear programming problem:

$$\begin{aligned}\text{Maximize:}\quad & g(x_1, x_2, \ldots, x_n) \\ & = g(x_{11}, \ldots, x_{1q}, x_{21}, \ldots, x_{2q}, \ldots, \\ & \qquad x_{n1}, \ldots, x_{nq}) \\ \text{Subject to:}\quad & x_i \in \chi, \qquad i = 1, 2, \ldots, n.\end{aligned}$$

Standard nonlinear programming approaches, such as steepest ascent, consider simultaneous changes in all nq coordinates. Exchange algorithms consider exchanges involving only the q coordinates associated with a given point. Computational efficiencies are exploited in exchange algorithms because the covariance matrix D, which is necessary for computation of the delta and deletion functions, is easily updated for exchanges of this type. In this framework, m-point exchanges involve changes to mq coordinates. Johnson and Nachtsheim (1983) showed empirically that multiple-point exchanges are not as effective as single-point exchanges. In fact, performance tended to improve with decreasing m. This suggests that there may be efficiencies associated with the exchange of a subset of the coordinates of a design point within an exchange algorithm. Stated differently, we motivate our approach by noting that the literature suggests that the performance of design algorithms seems to improve with level of greediness. This is the general idea behind the popular use of greedy algorithms in zero–one integer programming. See, for instance, Fox and Nachtsheim (1990) for a discussion. Relatedly, a greedy approach for constructing large-sample (approximate) optimal designs on finite design spaces was described by Robertazzi and Schwartz (1989).

Our purpose here is to investigate the performance of exact algorithms that exchange, where possible, only one factor-level at a time. For continuous factors, this will typically mean that only a single coordinate is changed; for discrete factors, changing an a-level factor will result in a simultaneous change of up to $a - 1$ coordinates. In our description of the algorithm, we assume that the q coordinates associated with a point $x \in \chi$ are partitioned into g subgroups, $x = (\tilde{x}_1', \ldots, \tilde{x}_g')'$, where the ith subgroup has q_i coordinates so that $q = \sum_{i=1}^{g} q_i$. We also assume that the design space χ can be written $\chi = \chi_1 \otimes \chi_2 \otimes \cdots \otimes \chi_g$ in correspondence with the preceding partition. For example, consider a three-factor experiment in which the first two factors are quantitative factors with operating ranges $[-1, 1]$ and $[5, 10]$ and the third is a four-level qualitative factor. Here we take $\chi = (x_1, x_2, x_3, x_4, x_5)' = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$, where $\tilde{x}_1 = x_1$, $\tilde{x}_2 = x_2$, and $\tilde{x}_3 = (x_3, x_4, x_5)'$. Hence $g = 3$, $q_1 = 1$, $q_2 = 1$, and $q_3 = 3$. Moreover, $\chi_1 = [-1, 1]$, $\chi_2 = [5, 10]$, and $\chi_3 = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\}$. In general, we shall try to make the individual q_i's as small as possible in an effort to minimize the computing associated with an exchange.

We note that if $q_i = 1$ for all i, a *cyclic coordinate–descent* algorithm results (Luenberger 1973). For instance, in the *Gauss–Southwell method*, the order in which coordinates are selected for descent is determined on the basis of the magnitudes of the gradient vector. In our implementation, coordinates are selected on a design-point-by-design-point basis as indicated by the deletion function. This preserves the ability to update the covariance matrix D cheaply via (6).

The approach described is not applicable, without modification, to candidate sets that cannot be written as factorial regions. For example, optimal design algorithms are frequently used to construct multiply constrained mixture designs. Mixture design spaces cannot be written as factorial regions. In such cases, we set $g = 1$, and the standard k-exchange algorithm results. When a subset of the factors is involved in a constraint, the corresponding coordinates can be identified as a coordinate group and the algorithm can be applied. An example of this approach is given in Section 4.

2.2 Coordinate-Exchange Algorithm for D Optimality

As with the k-exchange algorithm, at the start of an iteration, we identify the kq coordinates associated with the k least critical points in the current design via (2). These points are then exchanged on a coordinate-group-by-coordinate-group basis. Suppose that an exchange of the jth coordinate group of the ith point x_i is needed. Then $\tilde{x}_{ij}$ is to be replaced by $\tilde{x}_j \in \chi_j$ and the remaining coordinates of x_i are fixed. Let $\tilde{x}_{i,-j}$ denote the $q - q_j$ vector of coordinates that remain fixed during the exchange. Let $\Delta^{ij}(\tilde{x}_{ij}, \tilde{x}_j, \tilde{x}_{i,-j})$ represent the value of a delta when $\tilde{x}_{ij}$ is exchanged for $\tilde{x}_j$ and $\tilde{x}_{i,-j}$ remains fixed. Reordering terms as necessary, we partition $f(x)$ as

$$f(x) = \begin{pmatrix} f_1(\tilde{x}_j) \\ f_2(\tilde{x}_{-j}) \end{pmatrix}, \tag{7}$$

where $f_2(\tilde{x}_{-j})$ corresponds to the components of $f(x)$ that do not explicitly involve $\tilde{x}_j$. $f_1(\tilde{x}_j)$ and $f_2(\tilde{x}_{-j})$ are, respectively, p_1 and p_2 vectors, where $p_1 + p_2 = p$. For example, if $f'(x) = (1, x_1, x_2, x_1x_2, x_1^2)$ and $j = 1$, then $f_1'(\tilde{x}_1) = (x_1, x_1x_2, x_1^2)$ and $f_2'(\tilde{x}_{-1}) = (1, x_2)$, and in this case $p_1 = 3$ and $p_2 = 2$. Let $A = [1 - v(x_i)]D + Df(x_i)f'(x_i)D$, and partition A in correspondence with f:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}. \tag{8}$$

It is straightforward to show (Appendix B) that the delta function for D optimality can be written

$$\Delta_D^{ij}(\tilde{x}_{ij}, \tilde{x}_j, \tilde{x}_{i,-j}) = f_1'(\tilde{x}_j)A_{11}f_1(\tilde{x}_j) + a'f_1(\tilde{x}_j) + c, \tag{9}$$

where $a = 2A_{12}f_2'(\tilde{x}_{i,-j})$ and $c = f_2'(\tilde{x}_{i,-j})A_{22}f_2(\tilde{x}_{i,-j}) + (1 - v(x_i))$ are constant. Thus for D-optimality at the ith stage of an iteration we find

$$\max_{\tilde{x}_j \in \chi_j} \Delta_D^{ij}(\tilde{x}_{ij}, \tilde{x}_j, \tilde{x}_{i,-j}) = \Delta_D^{ij}(x_{ij}, \tilde{x}_j^*, \tilde{x}_{i,-j}).$$

Note that the maximization just indicated involves repeated evaluation of a quadratic form involving the $p_1 \times p_1$ matrix A_{11} and an inner product of two p_1 vectors. For example, for first-order models, both A_{11} and a are scalars; hence evaluation of the relevant part of Δ involves only four multiplications and an addition. This is in contrast to the point exchanges for convex design spaces, which require the evaluation of a quadratic form involving a $p \times p$ matrix. This simplification can lead to considerable savings when p is large.

Once $\tilde{x}_j^*$ has been determined, $\tilde{x}_{ij}$ is replaced by $\tilde{x}_j^*$, the objective function and the D matrix are updated, and the next coordinate group is considered. When all kq coordinates have been considered, the iteration ends and a termination check is performed. If sufficient improvements have been made, k new points are identified for deletion and the cycle continues. (In our implementation, the algorithm terminates when the relative change in the criterion drops below 10^{-4}.)

2.3 Cyclic Coordinate Exchanges for Linear Optimality

For linear optimality criteria of the form $L(D) = \text{tr}(DM)$ for M a positive semidefinite $p \times p$ matrix, an analogous algorithm can be given. The deletion function (5) is first used to identify k points in the current design for deletion. These k points are then sequentially exchanged for new points, as identified by optimization of the delta function for linear optimality. Again assume the partitioning of f as in (7), and let $B = \{B_{ij}\} = [1 - v(x_i)][I_p + 2Df(x)f'(x)]DWD - \phi(x_i)D$. Δ_L can then be expressed (Appendix B) as

$$\Delta_L^{ij}(\tilde{x}_{ij}, \tilde{x}_j, \tilde{x}_{i,-j}) = \frac{f_1'(\tilde{x}_j)B_{11}f_1(\tilde{x}_j) + b'f_1(\tilde{x}_j) + d}{f_1'(\tilde{x}_j)A_{11}f_1(\tilde{x}_j) + a'f_1(\tilde{x}_j) + c}, \tag{10}$$

where A_{11}, a, and c are as given near (9) and $b = 2B_{12}f_2(\tilde{x}_{i,-j})$ and $d = f_2'(\tilde{x}_{i,-j})B_{22}f_2(\tilde{x}_{i,-j})$ are constants. Thus the coordinate-exchange algorithm for linear optimality is essentially the same as that for D optimality; the only differences concern the specific form of the deletion and delta functions. A technical summary of the algorithm for both D and L optimality is given in Appendix C.

2.4 Optimizing the Delta Function in Convex Spaces

Optimization of the delta function can often be a difficult task. For finite design spaces, the procedure is conceptually simple, although the computational demands can be prohibitive when q is large. For convex (quantitative) design spaces, the tendency of hill-climbing routines to converge to local optima can hinder the effectiveness of the exchange procedure. These difficulties are largely avoided when a single factor level is exchanged within an exchange algorithm. In particular, certain advantages result if only a single coordinate of a design point is exchanged. For example, for regularly constrained convex design spaces, the global optimization of the delta function is possible by a grid search. The accuracy with which the optimum is determined will, of course, depend on the type of grid used.

Suppose that a single coordinate x_j is to be optimized, where $x_j \in \chi_j = [x_{\min}, x_{\max}]$. Let n_g^j denote the number of grid points to be equally spaced on the interval $[x_{\min}, x_{\max}]$. Since the exact D-optimal design for a first-order model consists entirely of vertices, the appropriate grid size is $n_g^j = 2$. For second- and higher-order models, we often use a telescoping sequence of grids. First, a coarse grid of size $n_g^j = 21$ is initially defined to locate the approximate optimal coordinate x_a. Then a new grid search is performed on the interval $[x_1, x_h]$, where $x_h = \min(x_a + \delta, x_{\max})$, $x_1 = \max(x_a - \delta, x_{\min})$, and $\delta = (x_{\min} - x_{\max})/(n_g^j - 1)$. Of course, line-search procedures could be used. An important advantage of the grid search is its robustness to premature termination at local optima.

2.5 Constructing Nonsingular Starting Designs

A cyclic coordinate augmentation procedure can be used for constructing starting designs or augmenting existing designs. Let X_i be the $i \times p$ design matrix consisting of i points. Galil and Kiefer (1980) suggested that the ith point, x_i, $1 \le i \le p$, be chosen to maximize $|X_i X_i'|$. For $i = 1$, $|X_1 X_1'| = f'(x_1)f(x_1)$. Using standard results in matrix algebra, one can show that, for $1 < i \le p$, x_i maximizes the *augmentation function*

$$a(x_i) = f'(x_i)[I - X_{i-1}'(X_{i-1}X_{i-1}')^{-1}X_{i-1}]f(x_i), \tag{11}$$

where I is the $p \times p$ identity matrix. For $p < i \le n$, the augmentation will depend on the criterion. Let D_{i-1} be the covariance matrix for the design consisting of the first $i - 1$ points selected. For D optimality, as noted in Section 1, optimal (one-point) augmentation is accomplished by adding to the current design that point that maximizes

$$a_D(x_i) = f'(x_i)D_{i-1}f(x_i). \tag{12}$$

For linear optimal designs, the best point to add maximizes

$$a_L(x_i) = \frac{\phi(x_i)}{1 - v(x_i)} \tag{13}$$

(Appendix B). Optimization of (11) and (12) or (13) on a coordinate-group-by-coordinate-group basis [as analogously described for delta functions (7) and (8)] is straightforward. During the ith augmentation, on optimizing the jth coordinate group, the point x_i can be partitioned as $x_i' = (\tilde{x}_{ij}, \tilde{x}_{i,-j}'^*, \tilde{x}_{i,-j}')$. Here, $\tilde{x}_{ij}$ are the coordinates to be optimized, $\tilde{x}_{i,-j}'^*$ are the fixed coordinates corresponding to the coordinate groups previously optimized, and $\tilde{x}_{i,-j}'$ are the fixed coordinates corresponding to coordinate groups not yet optimized. Note that $\tilde{x}_{i,-j}'$ must be set to evaluate the augmentation function. We suggest

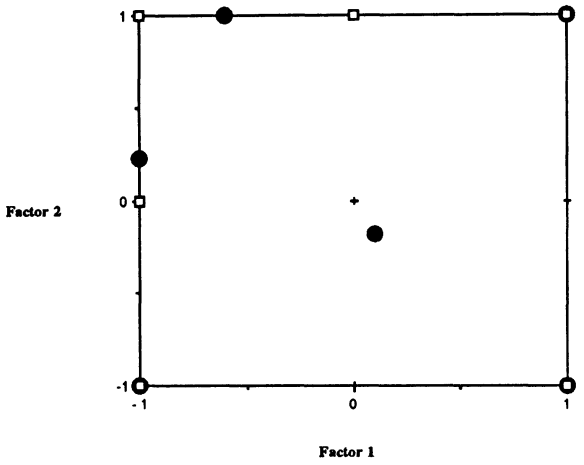

Figure 1. Optimal Design for Quadratic Regression, n = 6. The squares represent points of support of the optimal design computed by the k-exchange algorithm using a nine-point (3 × 3) candidate set. The solid dots are points of support resulting from the use of the coordinate-exchange algorithm.

that this be done by using coordinates associated with a point chosen randomly from χ. Alternatively, one could first add a point at random from χ, update D, and then reposition the point by optimizing the appropriate delta function. The former approach is simpler, requires fewer updates, and seems to perform comparably.

2.6 Illustration

As an illustration of the algorithm in the presence of a continuous design space, let $\chi = [-1, 1]^2$, let $f'(x) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$, and assume that $n = 6$. Because the design space is convex, it will sometimes be discretized as $C = \{-1, 0, 1\} \otimes \{-1, 0, 1\}$. D-optimal designs were constructed by the k-exchange algorithm using candidate set C and by the coordinate-exchange algorithm using χ with $n_g^j = 21, j = 1, 2$. The resulting designs are compared in Figure 1. Notice that the coordinate-exchange algorithm, in effect, reconfigured the three points in the second quadrant, moving them off the grid defined by C. Both the execution times and the D efficiencies of the designs were comparable. (The relative efficiency of the design found by the coordinate-exchange procedure was 1.0063.) This example underscores the fact that the coordinate-exchange algorithm effectively eliminates the need for standard nonlinear programming algorithms as used by Johnson and Nachtsheim (1983) and adjustment routines as suggested by Donev and Atkinson (1988).

3. PERFORMANCE

In this section, we study the performance of the cyclic coordinate-exchange algorithm for first- and second-order response-surface models for varying numbers of factors in the context of D optimality. For first-order models, we considered design problems having q factors for $4 \le q \le 13$ with $n = q + 1$ (saturated case) and $n = 2q$. The design space is $\chi = \chi_1 \otimes \chi_2 \otimes \cdots \otimes \chi_q$, where $\chi_i = [-1, 1]$. Because the model is first order, we take $n_g^i \equiv 2$ in the coordinate-exchange algorithm. For purposes of comparison, we also used the k-exchange algorithm in connection with a candidate space consisting of the 2^q vertices of χ. The number of points deleted at each iteration was set at $k = n$ for $n = q + 1$, and $k = q$ for $n = 2q$, for both algorithms. Starting designs were constructed in each case using the Galil–Kiefer algorithm as modified in Section 2. Although it is common practice to run a design algorithm from several starting designs, choosing the best resulting design, each test case in the current study was run just once. As suggested by an anonymous referee, the k-exchange algorithm was modified to incorporate the time-saving methods of Galil and Kiefer (1980). The results are presented in Table 1. For each problem solved by the coordinate-exchange algorithm, we report $|D|$ and the required execution time. (All runs were performed on a Cray-2 at the University of Minnesota.) For each problem solved by the k-exchange algorithm, we report the *relative* execution time (r_t) and the relative efficiency (r_e) of the final design. Specifically, r_t is the ratio of the execution time required by the cyclic coordinate-exchange algorithm to the time required by k exchange. r_e is defined as follows: Let D_c and D_k be the variance–covariance matrices of the designs constructed by the cyclic coordinate-exchange and k-exchange algorithms, respectively. Then

$$r_e = \left(\frac{|D_k|}{|D_c|}\right)^{1/p}. \tag{14}$$

The performances of the two algorithms are essentially indistinguishable on the basis of the D efficiencies of the designs produced. The average relative efficiency r_e for the 20 test cases is 1.008. In 5 of the 20 test cases, the k-exchange algorithm produced better designs than the coordinate-exchange algorithm; in 10 of the 20 cases, the reverse was true. Yet the time required by the k-exchange algorithm was substantially less. For large q, as indicated in Table 1, required time has been reduced by two orders of magnitude. As one might expect, execution time required by k exchange increases exponentially with q (i.e., linearly in the number of candidate points). Interestingly, the time required by the coordinate-exchange algorithm also increases exponentially with q but at a much slower rate. These relationships are depicted graphically in Figure 2.

Results for second-order response-surface models are summarized in Table 2. We considered design problems having q factors for $2 \le q \le 7$ with $n = p$ (saturated case) and $n \approx 1.5p$. (For the full second-order models, $p = 2q + \binom{q}{2} + 1$.) k was set to n for saturated designs and to approximately $n/4$ for the remaining cases. The design space is again $\chi = \chi_1 \otimes \chi_2 \otimes \cdots \otimes \chi_q$, where $\chi_i = [-1, 1]$. Because the model is second order, we take $n_g^i \equiv 3$ in the coordinate-exchange algorithm. For the k-exchange algorithm, the candidate set consisted of the

Table 1. Performance of Cyclic Coordinate Exchange for First-Order Models

Design problem				Performance of coordinate exchange		Performance relative to standard k exchange	
q	p	n	k	$\|D\|$	t	r_t	r_e
4	5	5	5	.4340E−3	.03	.451	1.0000
4	5	8	4	.4069E−4	.03	1.044	1.0000
5	6	6	6	.3906E−4	.06	.350	1.0000
5	6	10	5	.1246E−5	.06	.727	1.0000
6	7	7	7	.3014E−5	.07	.157	1.0342
6	7	12	6	.2791E−7	.10	.501	1.0607
7	8	8	8	.5960E−7	.13	.391	1.0000
7	8	14	7	.9313E−9	.15	.250	.9902
8	9	9	9	.4866E−8	.19	.047	1.0348
8	9	16	8	.1455E−10	.37	.187	1.0168
9	10	10	10	.2328E−9	.42	.031	1.0199
9	10	18	9	.3823E−12	.55	.113	.9885
10	11	11	11	.9313E−11	.62	.017	1.0356
10	11	20	10	.6121E−14	1.26	.077	.9899
11	12	12	12	.3725E−12	.90	.041	1.0158
11	12	22	11	.1012E−15	1.47	.008	1.0072
12	13	13	13	.8342E−14	1.74	.029	.9534
12	13	24	12	.1353E−17	2.53	.021	1.0066
13	14	14	14	.3232E−15	4.03	.026	.9948
13	14	26	13	.2062E−19	4.08	.013	1.0063

3^q points of $\chi = (-1, 0, 1)^q$. Results obtained are analogous to those for first-order models. In 3 of the 12 test cases, the designs found by the two algorithms were essentially the same. In four cases, the k-exchange algorithm found better designs, and in five cases the reverse was true. Significant reductions in computing time, however, were realized by the coordinate-exchange procedure. As indicated in Figure 3, execution time again appears to increase exponentially with increasing q for both algorithms; however, the rate of growth is much more manageable for the coordinate-exchange algorithm. With $q = 7$ and $n = 54$, it achieves a 27-fold reduction in required computing.

4. EXAMPLES

In this section we illustrate use of the algorithm in connection with two quality-improvement experiments. The first demonstrates its applicability to problems having many factors. The second demonstrates use of the algorithm in connection with an irregular design space. In both cases, coordinate groups involving more than a single coordinate are required. Moreover, both cases demonstrate use of the algorithm in connection with mixed qualitative-by-quantitative design spaces.

The following design problem was described by Bedford (1987). A 32-run designed experiment was to be

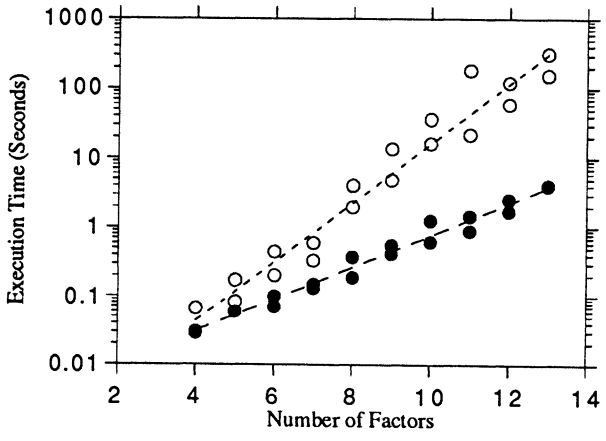

Figure 2. Comparison of Execution Times of First-Order Models: –•–, Coordinate Exchange; –o–, k Exchange with Candidate Set.

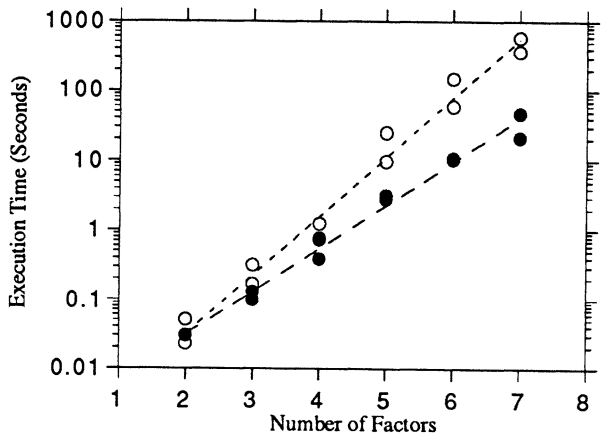

Figure 3. Comparison of Execution Times for Second-Order Models: –•–, Coordinate Exchange; –o–, k Exchange with Candidate Set.

Table 2. Performance of Cyclic Coordinate Exchange for Second-Order Models

Design problem				Performance of coordinate exchange		Performance relative to standard k exchange	
q	p	n	k	$\lvert D\rvert$	t	r_t	r_e
2	6	6	6	.3906E−2	.03	.590	1.0000
2	6	9	3	.1929E−3	.03	1.318	1.0000
3	10	10	10	.3815E−5	.10	.317	.8941
3	10	15	5	.4147E−8	.13	.770	.9997
4	15	15	15	.2874E−11	.39	.499	1.0000
4	15	22	7	.6667E−15	.73	.587	1.0076
5	21	21	21	.2168E−20	3.22	.130	1.0147
5	21	32	8	.1354E−24	2.80	.286	1.0012
6	28	28	28	.1146E−30	10.78	.072	1.0761
6	28	42	8	.1617E−36	10.48	.177	.9915
7	36	36	36	.7757E−43	47.14	.128	1.1091
7	36	54	13	.2215E−51	21.29	.037	.9897

carried out in an attempt to reduce the rework rate of a diesel injector. The quality characteristic of interest was the nozzle-opening pressure. Brainstorming suggested the 16 factors and associated levels given in Table 3. Although the four-level factors are clearly quantitative, given the irregular spacing, we presume that the experimenter wanted each of these levels tested. Thus, for purposes of this example, we treat these factors as qualitative. All of the remaining qualitative factors were at two levels. This led to the 16 coordinate groups also summarized in Table 3. The marginal spaces, $\chi_i, i = 1, 2, \ldots, 16$, are also given in the table. (For qualitative factors, we assumed an "indicator" parameterization; others could be used.) A candidate set for this problem would consist of $4^2 \times 3 \times 2^{13} = 393,216$ points, which *precludes* use of standard exchange algorithms. Assuming a main-effects model, we have $p = 22$. The cyclic coordinate-exchange algorithm was used to search for a D-optimal design for $n = 32$, using $k = 8$. Forty-one seconds were required for convergence (again using the Cray-2 at the University of Minnesota Supercomputer Institute), and the design constructed was not as efficient as the orthogonal-array design used by the authors. Moreover, when we used the orthogonal-array design as a starting design, the cyclic coordinate-exchange algorithm was not able to improve on it. This suggests that the orthogonal-array design is an exact D-optimal design. An advantage of the optimal-design approach, of course, is that the experimenter need

Table 3. Factors and Coordinate Groups for Example 1

Factor	Type	No. of levels	Levels	Coordinate group		
				i	q_i	χ_i
Pump deliver rate	Quant	2	3 turns in, 3 turns out	1	1	[−3, 3]
Air pressure	Quant	2	3.5 ATS, 4.5 Ats	2	1	[3.5, 4.5]
Test fixture type	Qual	2	One pin, two pin	3	1	{1, −1}
Spring compression	Quant	2	1 mm, 1.8 mm	4	1	[1, 1.8]
Spring scragg	Qual	2	On, off	5	1	{−1, 1}
Position in fixture	Qual	2	Vertical, horizontal	6	1	{−1, 1}
Oil cleanliness	Qual	2	New, old	7	1	{−1, 1}
Cone material	Qual	2	Steel, plastic	8	1	{−1, 1}
Cap nut torque	Quant	2	35, 50	9	1	[35, 50]
Preclamp	Qual	2	Incorrect, correct	10	1	{−1, 1}
Injector tap	Qual	2	Yes, no	11	1	{−1, 1}
Clamping pressure	Qual	4	500, 750, 1,000, 2,000	12	3	χ^*_{12}
Nominal shim thickness	Quant	3	1, 1.28, 1.56	13	1	[1, 1.56]
Purge time	Qual	4	0, 1, 5, 15 (seconds)	14	3	χ^*_{14}
Delay time 1	Quant	2	5 flashes, 20 flashes	15	1	[5, 20]
Delay time 2	Quant	2	5 seconds, 10 seconds	16	1	[5, 10]

*$\chi_{12} = \chi_{14} = \{(1,0,0), (0,1,0), (0,0,1), (0,0,0)\}$.

Table 4. Factors and Coordinate Groups for Example 2

Factor	Type	No. of levels	Levels	Coordinate group i	q_i	χ_i
Wire	Qual	2	Silicone bronze, nickel iron	1	2	χ_1^a
Heat	Quant	3	275, 325, 360 (silicone wire) 200, 225, 275 (nickel wire)			(χ_1^a)
Drum	Qual	3	cold, preheat, grind	2	2	χ_2^b
Per flow time	Quant	3	0, .5, 1	3	1	[0, 1]
Post flow time	Quant	3	0, .5, 3	4	1	[0, 3]
Wire feed speed	Quant	3	40, 60, 70	5	1	[40, 70]
Weld time	Quant	3	1, 1.5, 2	6	1	[1, 2]
Crater time	Quant	3	1.25, 1.75, 2.5	7	1	[1.25, 2.5]

[a] $\chi_1 = \{1\} \otimes [275, 360] \cup \{-1\} \otimes [200, 275]$, discretized as {(1, 275), (1, 317.5), (1, 360), (−1, 200), (−1,237.5), (−1, 275)}.
[b] $\chi_2 = \{(1,0), (0,1), (0,0)\}$.

not be constrained on $n = 32$. We also computed designs for $n = 22$ (saturated case), 24, and 28. The normalized relative efficiencies were 87%, 90%, and 99% when compared to the 32-point design, and the computing times were 22, 31, and 57 seconds, respectively.

Szelazek (1987) reported on the design and analysis of a quality-improvement experiment to optimize the welding of balance weights in the manufacture of light-truck brake drums. The main concern was the strength of the weld. An 18-run experiment was conducted, with factors and associated ranges as described in Table 4. The design space is irregular because of the "sliding scale" used with the second factor. The author used the L_{18} layout and analyzed the data under the mistaken assumption that the design was orthogonal. Because different ranges were used for heat depending on the type of wire, however, the levels of these first two factors are correlated. We use the coordinate-exchange algorithm to construct a (nearly) D-optimal 18-point main-effects design on the appropriate design space. The irregularity is handled combining the first two factors in a single coordinate group as described in Table 4. Note that at each level of wire (silicone or nickel) we have specified three equally spaced levels of heat in the coordinate group. Although this is a fairly standard approach, we could have used the (unequal) spacing as originally specified by the experimenter. The model under scrutiny is additive, including all main effects and quadratic effects for all factors but the first. Thus $p = 16$, and we search for a D-optimal design using $k = 4$. After a single try, the algorithm produced a solution design with relative efficiency of 1.08 in 9.4 seconds. The slight increase in D efficiency is not surprising because the irregularity of the design space was explicitly considered in the construction of the design. Moreover, the use of equal spacing for the levels of heat may have contributed to the efficiency of the design produced by the coordinate-exchange algorithm.

5. DISCUSSION

We have described a simple, new approach to constructing exact optimal designs. As indicated in the introduction, the coordinate-exchange approach offers several advantages over existing, candidate-set-based procedures. Among the more important are the following: First, the algorithm can handle convex spaces, factorial candidate spaces, or mixtures of these. Because of its ability to handle convex spaces directly, it eliminates the need for standard nonlinear programming approaches when optimizing delta or augmentation functions. For discrete factors, candidate spaces are not constructed, removing a barrier to the use of optimal design in very large problems. Finally we reemphasize the ease with which the algorithm can be implemented and the speed with which it constructs highly efficient designs.

We have demonstrated the impact of coordinate exchanges in the context of the k-exchange algorithm; yet there is nothing about the approach that would preclude its application in connection with other well-known exchange algorithms such as DETMAX (Mitchell 1974). For example, the coordinatewise augmentation procedure discussed in Section 2.5 would be central to the development of a coordinate-exchange-based excursion of DETMAX. Although we have not attempted any such implementation, we would expect improvements analogous to those described in this article.

ACKNOWLEDGMENTS

We thank the editor and three referees for comments that led to substantive improvements in content and exposition. This research was funded in part by the Minnesota Supercomputer Institute and the McKnight Foundation. The second author thanks Friedrich Pukelsheim for his invitation to visit the Institut fur Mathematik, Universitaet Augsburg, where a part of this work was carried out.

APPENDIX A. DELETION AND AUGMENTATION FUNCTIONS FOR LINEAR OPTIMALITY

Let M, U, and V be $p \times p$, $p \times q$, and $q \times p$ matrices, respectively. For M nonsingular, we have (e.g, see Searle 1982, p. 153)

$$[M + UV]^{-1} = M^{-1} - M^{-1}U\left[I_q + VM^{-1}U\right]^{-1}M^{-1}. \tag{A.1}$$

Let $U = \pm f(x)$ (+ for augmentation, − for deletion), $V = f'(x)$, and $M = X'X$. We have

$$D_{\text{up}} = [X'X \pm f(x)f'(x)]^{-1} = D \pm \frac{-Df(x)f'(x)D}{1 \pm v(x)}.$$

Evaluating L on both sides yields the results

$$d_L(x_i) = \frac{\phi(x_i)}{1 - v(x_i)}$$

and

$$a_L(x_i) = \frac{\phi(x_i)}{1 + v(x_i)}.$$

APPENDIX B. DERIVATION OF THE Δ FUNCTIONS FOR COORDINATE EXCHANGES

From (1), one can easily show that $\Delta_D(x_i, x) = f'(x)Af(x) + (1 - v(x_i))$, where $A = (1 - v(x_i))D + Df(x_i)f'(x_i)D$. Assuming the partitioning indicated in (7) and (8), we have $\Delta_D^{ij}(\tilde{x}_{ij}, \tilde{x}_j, \tilde{x}_{i,-j}) = f_1'(\tilde{x}_j) A_{11} f_1(\tilde{x}_j) + 2f_2'(\tilde{x}_{i,-j}) A_{21} f_1(\tilde{x}_j) + f_2'(\tilde{x}_{i,-j}) A_{22} f_2(\tilde{x}_{i,-j}) + (1 - v(x_i))$, from which (9) is immediate.

For $L(D) = \text{tr}[DM]$ with M positive semidefinite, from (4),

$$\Delta_L(x_i, x) = \frac{f'(x)Bf(x)}{\Delta_D(x_i, x)},$$

where $B = [1 - v(x_i)][I_p + 2Df(x_i)f'(x_i)]DWD - \phi(x_i)D$. Assuming the reordering (7) and a corresponding partitioning of B, the numerator in (10) follows immediately. The denominator follows from (9).

APPENDIX C. COORDINATE-EXCHANGE ALGORITHM

BEGIN
Define the g coordinate groups. Select k for use in the k-exchange algorithm and ε for termination. Obtain a nonsingular starting design using the algorithm described in Section 2.5. For linear optimality, compute the moment matrix, M. In what follows, D is the current dispersion matrix; $c(D)$ is the current objective function value [$c(D) = |D|$ for D optimality; $c(D) = \text{tr}[DM]$ for linear optimality.]

```
REPEAT
    c0 = c(D)
    for i = 1 to n
        compute d(x_i) via (2) or (5)
    end i
    Identify and rank the k points x^1, ..., x^k that have
    the smallest values of d(x_i)
    for i = 1 to k
        for j = 1 to g
            x̃_j^* = arg max_{x̃_j ∈ χ_j} Δ^{ij}(x̃_ij, x̃_j, x̃_{i,-j}) via (9)
            or (10)
            replace x̃_ij by x̃_j^*; update D via (6); update
            c(D) via (1) or (4)
        end j
    end i
UNTIL
    c0 − c(D) < ε
END
```

[*Received October 1991. Revised July 1994.*]

REFERENCES

Atkinson, A. C., and Donev, A. N. (1989), "The Construction of Exact D-optimum Experimental Designs With Application to Blocking Response Surface Designs," *Biometrika*, 76, 515–526.

Bedford, S. J. (1987), "Reduction in Rework of Diesel Injectors," in *Fifth Symposium on Taguchi Methods*, Dearborn, MI: ASI Press, pp. 589–605.

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

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

——— (1982), "An Analysis of the k-exchange Algorithm," technical report, University of Minnesota, Dept. of Operations and Management Science.

Donev, A. N., and Atkinson, A. C. (1988), "An Adjustment Algorithm for the Construction of Exact D-optimum Experimental Designs," *Technometrics*, 30, 429–433.

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

Fox, G. E., and Nachtsheim, C. J. (1990), "An Analysis of Six Greedy Selection Rules for Zero-one Integer Programming," *Naval Research Logistics*, 37, 299–307.

Galil, Z., and Kiefer, J. (1980), "Time- and Space-Saving Computer Methods, Related to Mitchell's DETMAX, for Finding D-Optimum Designs," *Technometrics*, 21, 301–313.

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

Johnson, M. E., and Nachtsheim, C. J., (1983), "Some Guidelines for Constructing Exact D-Optimal Designs on Convex Design Spaces," *Technometrics*, 25, 271–277.

Koch, D., Morris, W. T., Nachtsheim, C. J., and Welch, W. J. (1991), "Computer Software for Robust Product Design," unpublished manuscript.

Luenberger, D. G. (1973), *Introduction to Linear and Nonlinear Programming*, Reading, MA: Addison-Wesley.

Meyer, R. K., (1989), "Contributions to the Construction of Exact D-optimal Experimental Designs," unpublished Ph.D. thesis, University of Minnesota, Operations and Management Science Dept.

Meyer, R. K., and Nachtsheim, C. J. (1988), "Simulated Annealing in the Construction of Exact Optimal Design of Experiments," *American Journal of Mathematical and Management Science*, 8, 329–359.

Mitchell, T. J. (1974), "An Algorithm for the Construction of 'D-

Optimal Experimental Designs," *Technometrics*, 16, 203–210.

Nachtsheim, C. J. (1987), "Tools for Computer-Aided Design of Experiments," *Journal of Quality Technology*, 19, 132–160.

Powell, M. J. D., (1964), "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives," *Computer Journal*, 7, 155–162.

Robertazzi, T. G., and Schwartz, S. C., (1989), "An Accelerated Sequential Algorithm for Producing *D*-Optimal Designs," *SIAM Journal of Scientific and Statistical Computing*, 10, 341–358.

Searle, S. R. (1982), *Matrix Algebra Useful for Statistics*, New York: John Wiley.

Szelazek, G. B. (1987), "Welding Brake Drum Balance Weights," in *Fifth Symposium on Taguchi Methods*, Dearborn, MI: ASI Press, pp. 573–587.

Welch, W. J. (1982), "Branch-and-Bound Search for Experimental Designs Based on *D* Optimality and Other Criteria," *Technometrics*, 26, 217–224.

——— (1984), "Computer-Aided Design of Experiments for Response Estimation," *Technometrics*, 26, 217–224.

Wynn, H. P. (1972), "Results in the Theory and Construction of *D*-Optimum Experimental Designs," *Journal of the Royal Statistical Society*, Ser. B, 34, 133–147.