Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

An Exact Algorithm for Maximum Entropy Sampling

Chun-Wa Ko, Jon Lee, Maurice Queyranne,

To cite this article:
Chun-Wa Ko, Jon Lee, Maurice Queyranne, (1995) An Exact Algorithm for Maximum Entropy Sampling. Operations Research 43(4):684-691. http://dx.doi.org/10.1287/opre.43.4.684

AN EXACT ALGORITHM FOR MAXIMUM ENTROPY SAMPLING

CHUN-WA KO

Rutgers University, New Brunswick, New Jersey

JON LEE

University of Kentucky, Lexington, Kentucky

MAURICE QUEYRANNE

University of British Columbia, Vancouver, British Columbia, Canada

(Received January 1993; revision received September 1993; accepted April 1994)

We study the experimental design problem of selecting a most informative subset, having prespecified size, from a set of correlated random variables. The problem arises in many applied domains, such as meteorology, environmental statistics, and statistical geology. In these applications, observations can be collected at different locations, and possibly, at different times. Information is measured by "entropy." In the Gaussian case, the problem is recast as that of maximizing the determinant of the covariance matrix of the chosen subset. We demonstrate that this problem is NP-hard. We establish an upper bound for the entropy, based on the eigenvalue interlacing property, and we incorporate this bound in a branch-and-bound algorithm for the exact solution of the problem. We present computational results for estimated covariance matrices that correspond to sets of environmental monitoring stations in the United States.

We consider experimental situations in which we wish to make statistical inferences regarding a set of random variables from observations of a subset of these variables. In practice, the variables may be dispersed over space and/or time. For example, the random variables may correspond to potential observations of meteorological or environmental monitoring stations. Other examples occur in statistical geology where the observations may be collected at different points in space. Maintaining and operating all possible observation points or stations is costly, and one may want to select only a subset of them. In such circumstances, it may also be required, for a variety of other scientific, historical, or political reasons, that certain specified points be included in the resulting subset. We study the problem of selecting a "best" such subset with specified size.

Formally, we are given a set N of n points, called the *design space*, a set $F \subset N$ of f "forced" points, and a design size s, such that $f \leq s \leq n$. Our goal is to choose a set S of s points satisfying $F \subset S \subset N$, called a *feasible design*, such that observations taken at these points will be as valuable as possible.

When the design space is finite and the criterion for evaluating designs is well defined, the design problem may be viewed as a combinatorial optimization problem. There is a rich body of literature on *combinatorial design*, where the design space and the criterion are highly structured combinatorial objects, for example, when the design space consists of all p-vectors with ± 1 entries. Here, we consider a somewhat less structured class of problems where both the design space and the criterion are derived from observed and/or estimated data, and our goal is to choose a feasible design that minimizes the resulting uncertainty. To measure this uncertainty, we associate with the design space N a symmetric positive definite $n \times n$ matrix A, for example, a covariance matrix. Then the *entropy* associated with any s-element subset S of N is the logarithm of the determinant of the $s \times s$ principal submatrix $A[S]$ with row and column indices in S. A *D_s-optimal design* is a feasible design that has maximum entropy. As the logarithm is an increasing function, it is an equivalent problem to maximize $\det(A[S])$ among the set of feasible designs.

Among the numerous applications of D_s-optimal designs (see, e.g., Mitchell 1974a, the papers collected in Dodge, Fedorov and Wynn 1988, and the references therein), the optimal design of spatial sampling networks has received considerable recent attention (e.g., Shewry and Wynn 1987, Fedorov and Hackl 1990, the references therein, and those below). A spatial sampling network consists of a set S of s points, called *stations*, dispersed in a (usually two-dimensional) space and used to measure some variable of interest. The matrix $A[S]$ defines the correlation between the observations at the various stations. Our interest in the optimum design of spatial sampling networks is motivated by the ongoing work of Professor J. V. Zidek (Department of Statistics, University of British Columbia) and his associates, on the evaluation, design,

Subject classifications: Facilities/equipment planning: location of environmental monitoring stations. Programming: nonlinear combinatorial optimization. Statistics: maximum entropy sampling.

Area of review: OPTIMIZATION.

0030-364X/95/4304-0684 $01.25

and redesign of environmental monitoring networks. The stations in these networks measure such variables as acidic deposition in precipitation over large geographical areas. The spatial covariance matrix A reflects such effects as wind patterns, mountains, and the general structure of precipitation fields. In the case of the contraction of an existing network (Wu and Zidek 1992), the n candidate points represent an existing network from which the s surviving stations have to be selected, and the matrix A is estimated from historical observations at the n existing stations. In the case of network expansion, a number of candidate sites are selected, for example, grid points over the region of interest, and/or sites that are particularly attractive (e.g., existing stations in other related networks). The spatial covariance matrix A is then estimated in part from historical observations, and in part by interpolation and/or extrapolation of available data using appropriate models; see Guttorp et al. (1992) for details.

The purpose of this paper is threefold. First, we investigate the computational complexity of our problem. Then we develop an exact algorithm for finding a D_s-optimal design. Finally, we report on the application of this algorithm to a number of problems with real data that arise from studies of the redesign of environmental monitoring networks.

After formalizing our definitions and notation in Section 1, we consider in Section 2 the computational complexity of various problems related to optimizing the function f: $2^N \to \Re$ defined by $f(S) := \log_2(\det(A[S]))$. Kelmans and Kimelfeld (1983) show that this is a *submodular* set function, that is,

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T) \quad \text{for all } S, T \subseteq N.$$

Although submodularity is a special combinatorial property (e.g., Nemhauser and Wolsey 1981, 1988) it is not known to imply that a polynomial-time algorithm exists for our problem. Indeed, we show that our problem is NP-hard, implying that it is unlikely that an efficient algorithm can be found to optimally solve all instances of our problem. The methods used by statisticians for finding D_s-optimal designs consist of complete enumeration and heuristics, based mostly on exchanges. The best known of these is probably the DETMAX method of Mitchell (1974a, b); also see the above references for related methods. A greedy constructive algorithm is sometimes used to construct an initial solution (e.g., Guttorp et al.). Our computational results partly vindicate the recourse to such heuristics. Indeed, it is well known (see Nemhauser and Wolsey) that greedy and exchange heuristics often produce very good solutions to problems of maximizing submodular functions.

However, because of the importance of the observations and budgets related to spatial sampling networks, and to avoid potential uncertainties in evaluating design methodologies which would accrue from using suboptimum solutions, it is desirable to have robust methods for producing truly optimal solutions for instances of moderate size. Complete enumeration may be used when n and s are not too large. For example, Guttorp et al. used it to optimally select 8 of 16 potential sites to add to a current 36-station network. When trying to select 13 of 27 sites to be added to the same network, they found an initial solution using a greedy algorithm. However, they "have not yet been able to compare it with the true optimum because our [complete enumeration algorithm] has not found the optimum after several weeks of running time"! In Section 3, we use the interlacing properties of eigenvalues to provide an upper bound on the value of a D_s-optimal design. We use this upper bound in Section 4 to develop an exact, branch-and-bound algorithm for finding a D_s-optimal design.

In Section 5, we report on computational experiments with this exact algorithm, using real data from a study of environmental monitoring networks. For example, we find and verify the optimality of the greedy solution to Guttorp et al.'s problem of selecting 13 of 27 additional sites, using 7 seconds on a modest workstation.

1. DEFINITIONS AND NOTATION

Let $\mathcal{N} = (1, 2, \ldots, n)$ and $N = \{1, 2, \ldots, n\}$, where n is a positive integer. Throughout, A denotes a real symmetric positive definite matrix with rows and columns indexed by $\mathcal{N}$. Hence $A_{ij} = A_{ji}$ for all $i, j \in N$, and $x'Ax > 0$ for all $x \in \mathbf{R}^n$. For an s-element subset S of N $(1 \leq s \leq n)$, $\mathcal{S}$ denotes the s-tuple of elements of S taken in increasing order. Let $A[S]$ denote the principal submatrix of A having rows and columns indexed by $\mathcal{S}$. Similarly, we write $A[S_1, S_2]$ to denote the submatrix of A having rows indexed by $\mathcal{S}_1$ and columns indexed by $\mathcal{S}_2$. We note that $A[S, S] = A[S]$, and the symmetry of A implies that the transpose of $A[S_1, S_2]$ is $A[S_2, S_1]$. Throughout, we omit the braces around the element of single-element sets. Hence, $A[i, j]$ denotes A_{ij}. We write $\det(A[S])$ to denote the determinant of $A[S]$, and $\text{spec}(A[S])$ to denote the multiset of eigenvalues of $A[S]$. Hence, $\text{spec}(A[S])$ has s (not necessarily distinct) elements. The properties of A imply that $\det(A[S])$ is always nonnegative, and that the elements of $\text{spec}(A[S])$ are nonnegative real numbers. Finally, let $\lambda_i(A[S])$ denote the ith greatest eigenvalue of $A[S]$ $(1 \leq i \leq s)$.

Let F be an f-element subset of N $(0 \leq f \leq n)$. Let E be an e-element subset of N $(0 \leq e \leq n - f)$ satisfying $F \cap E = \emptyset$. Let s be a positive integer satisfying $f \leq s \leq f + e$. Our optimization problem $\mathbf{P}(\max, \|)$ is to determine

$$v(A, F, E, s) = \max_{\substack{S:|S|=s, \\ F \subset S \subset F \cup E}} \det(A[S]),$$

and an associated maximizing S. Any s-element subset S of N satisfying $F \subset S \subset F \cup E$ is called a *feasible solution*. The set F consists of the indices that are *forced* into every feasible solution. The set E consists of the

indices that are *eligible* for consideration. There is really no loss of generality in taking $N = F \cup E$ and replacing A with $A[F \cup E]$, but casting the problem as we have will ease the exposition of the algorithm.

2. COMPLEXITY

We now study the computational complexity of problem **P**(max, ||) and of the closely related problems **P**(min, ||) (cardinality-constrained minimization), **P**(max, ·) (unconstrained maximization) and **P**(min, ·) (unconstrained minimization). For this, we assume that matrix A has rational entries, each expressed as a pair of (relatively prime) integers. Where the problems include a cardinality constraint $|S| = s$, we also consider the number s as part of the input. Thus, the *input size* is the total number of bits required to input the matrix A, a bit vector specifying which row/column indices of A are in F, and possibly the number s; see Garey and Johnson (1979) and Grötschel, Lovász and Schrijver (1988).

Theorem 1.

i. *The problem* **P**(*min*, ·) *is solvable in polynomial time.*
ii. *The problem* **P**(*min*, ||) *is NP-hard (even if A only has entries in* $\{0, 1, 3n\}$ *and* $F = \emptyset$).
iii. *The problem* **P**(*max*, ·) *is NP-hard (even if* $F = \emptyset$).
iv. *The problem* **P**(*max*, ||) *is NP-hard (even if A only has entries in* $\{0, 1, 3n\}$ *and* $F = \emptyset$).

Before getting to the proof of Theorem 1, we establish some lemmata.

Lemma 1. *Given positive integers* $s \leq n$, *let* B *and* B' *be* $s \times s$ *symmetric positive definite matrices with all diagonal entries equal to* $3n$, *and all off-diagonal entries equal to* 0 *or* 1. *Assume that* $B[i, j] \leq B'[i, j]$ *for all* i, j, *and that at least one of these inequalities is strict. Then* $det(B) > det(B')$.

Proof. We only need to prove Lemma 1 for the case where B and B' differ in only two entries, say (without loss of generality) $B[1, 2] = B[2, 1] = 0$ and $B'[1, 2] = B'[2, 1] = 1$. Let $B_1 = B[\{1, 2\}]$, $B'_1 = B'[\{1, 2\}]$, $B_2 = B[\{1, 2\}, \{3, \ldots, n\}]$, and $B_3 = B[\{3, \ldots, n\}]$. Then

$$\begin{pmatrix} B_1 & B_2 \\ B_2^t & B_3 \end{pmatrix} = \begin{pmatrix} I & B_2 \\ 0 & B_3 \end{pmatrix} \begin{pmatrix} B_1 - C & 0 \\ B_3^{-1}B_2^t & I \end{pmatrix},$$

where $C = (c_{ij})$ denotes the matrix $B_2B_3^{-1}B_2^t$. Thus, $\det(B) = \det(B_3)\det(B_1 - C)$, and, similarly, $\det(B') = \det(B_3)\det(B'_1 - C)$. Since $\det(B_1 - C) = (3n - c_{11})(3n - c_{22}) - c_{12}^2$, and $\det(B'_1 - C) = (3n - c_{11})(3n - c_{22}) - (1 - c_{12})^2$, it suffices to show that $c_{12} < 1/2$. To that end, observe that B_3^{-1} is real, symmetric, and positive definite. Thus, $\|B_3^{-1}\|_2 = \lambda_{max}(B_3^{-1}) = 1/\lambda_{min}(B_3)$. All eigenvalues of B_3 are at least $3n - (n - 3) = 2n + 3$ by the Geršgorin Disc Theorem. Hence, with the two columns of B_2^t denoted by b_1 and b_2,

$$\begin{aligned} |c_{12}| &= |b_1^t B_3^{-1} b_2| \\ &\leq \|b_1\|_2 \|b_2\|_2 \|B_3^{-1}\|_2 \\ &\leq (n - 2)/(2n + 3) \end{aligned}$$

which is less than 1/2 for any positive integer n. This completes the proof.

Let G be a simple undirected graph with vertex set $\mathcal{V}(G) = N$ and edge set $\mathcal{E}(G)$. We define the $n \times n$ matrix A_G by

$$A_G[i, j] := \begin{cases} 3n & \text{if } i = j; \\ 1 & \text{if } \{i, j\} \in \mathcal{E}(G); \\ 0 & \text{otherwise.} \end{cases}$$

Note that A_G is symmetric and diagonally dominant; hence it is positive definite. For any subset $S \subseteq N$, the *induced subgraph* G_S is defined to be the graph having vertex set $\mathcal{V}(G_S) = S$ and edge set $\mathcal{E}(G_S) = \mathcal{E}(G) \cap (\mathcal{V}(G_S) \times \mathcal{V}(G_S))$. A *clique* (respectively, *stable set*) in G is a subset $S \subseteq N$ such that G_S is a complete graph (respectively, a graph with no edges).

Lemma 2. *Let* S *be an* s*-element subset of the vertex set* $\mathcal{V}(G)$ *of graph* G, *and let* $n = |\mathcal{V}(G)|$. *Then we have*

$$(3n + s - 1)(3n - 1)^{s-1} \leq \det(A_G[S]) \leq (3n)^s,$$

where the first (respectively, second) inequality holds as an equation if and only if S *is a clique (resp., a stable set) in* G.

Proof. By Lemma 1, it suffices to check that if S is a stable set of G, then $\det(A_G[S]) = (3n)^s$, and if S is a clique of G, then $\det(A_G[S]) = (3n + s - 1)(3n - 1)^{s-1}$. The first part of the statement is obvious. The second part follows, in the following manner, from well-known properties of circulant matrices: By Theorem 3.2.2 of Davis (1979), if S is a clique, then the eigenvalues of $(A_G[S])$ are

$$\lambda_j := 3n + \sum_{l=1}^{s-1} \omega^{l(j-1)} \qquad (1 \leq j \leq s),$$

where ω is any primitive sth root of unity, for example $2\pi i/s$. Therefore, $\lambda_1 = 3n + s - 1$. Also, by using the standard formula to sum a finite geometric series and the fact that ω is an sth root of unity, we have $\lambda_j = 3n - 1$ for $2 \leq j \leq s$. The result follows.

The following result refines the second inequality in Lemma 2.

Lemma 3. *If an* s*-element subset* S *of the vertex set* N *of the* n*-vertex graph* G *does not induce a stable set, then*

$$\det(A_G[S]) \leq (1 - (3n)^{-2})(3n)^s.$$

Proof. By Lemma 1, it suffices to prove this inequality for the case where S induces a subgraph with a single

edge. The result easily follows by expanding the determinant of $A_G[S]$ along the column indexed by either endpoint of the single edge.

Proof of Theorem 1. First recall that the determinant of a rational matrix can be computed in time polynomial in the input size (Edmonds 1967). Hence the decision versions (Garey and Johnson) of all problems in i–iv are in NP.

As indicated in the Introduction, the function $f: 2^N \to \Re$ defined by $f(S) := \log_2(\det(A[S]))$ is submodular (Kelmans and Kimelfeld), and for any subset S, the value $f(S)$ can be approximated in polynomial time to any desired precision. Furthermore, the function $f_F: 2^{N\setminus F} \to \Re$ defined by $f_F(S) := f(S \cup F) - f(F)$ is submodular. Thus, we can use the algorithm in Grötschel, Lovász and Schrijver to solve $P(\min, \cdot)$ in polynomial time. That is, part i is established.

Results ii and iv immediately follow from Lemma 2 by reduction from the well-known NP-hard problems CLIQUE and STABLE SET, respectively (see Garey and Johnson).

Finally, we establish part iii. Consider the NP-complete problem:

STABLE SET: *Given an n-vertex graph G and an integer $p < n$; decide whether G contains a stable set with p vertices.*

Pick a rational number α satisfying

$$(3n)^{-1} < \alpha < (3n)^{-1}(1 - (3n)^{-2})^{-1/(n-p)}.$$

Note that such a number exists and can be found in polynomial time, by binary search over the interval $(1/(3n), 2/(3n))$. Letting $A = \alpha A_G$, we claim that G contains a stable set with at least p vertices if and only if

$$\max\{\det(A[S]): \emptyset \subset S \subseteq V(G)\} \geq \alpha^p(3n)^p.$$

For this, first note that $\det(A[S]) = \alpha^s \det(A_G[S])$ for all s-vertex subsets S. If G contains a stable set S of size p, then $\det(A[S]) = \alpha^p(3n)^p$, thus proving the necessity of the above inequality. To prove sufficiency, assume that G contains no stable set of size p and consider any vertex subset S. Let $s := |S|$. If $s < p$ we have, from Lemma 2, $\det(A[S]) \leq \alpha^s(3n)^s < \alpha^p(3n)^p$. If $s \geq p$, then S cannot be a stable set, and using Lemma 3, we have

$$\begin{aligned}\det(A[S]) &\leq \alpha^s(3n)^s(1 - (3n)^{-2}) \\ &\leq \alpha^n(3n)^n(1 - (3n)^{-2}) < \alpha^p(3n)^p.\end{aligned}$$

This proves the sufficiency of the above condition. The proof of Theorem 1 is complete.

3. AN UPPER BOUND

In the present section, we establish an upper bound on $v(A, F, E, s)$. Any such upper bound is potentially useful in establishing the quality of a feasible solution. Moreover, a reasonably strong upper bound is a key ingredient in any implicit enumeration scheme. Our bound is a straightforward consequence of the well-known eigenvalue interlacing property for symmetric matrices. Another upper-bounding method follows from general linear/integer programming methods for maximizing a submodular function (see Nemhauser and Wolsey 1981).

Lemma 4. (see Wilkinson 1965, pp. 103–104). *Let B be a symmetric matrix with rows and columns indexed by $\mathcal{N}$. Let $R = \{1, 2, \ldots, r\}$ and let $B_r = B[R]$, for $1 \leq r \leq n - 1$. Then the interlacing property holds:*

$$\begin{aligned}&\lambda_{r+1}(B_{r+1}) \leq \lambda_r(B_r) \leq \lambda_r(B_{r+1}) \leq \cdots \leq \lambda_2(B_{r+1}) \\ &\quad \leq \lambda_1(B_r) \leq \lambda_1(B_{r+1}).\end{aligned}$$

Proposition

$$v(A, \emptyset, E, s) \leq b(A, \emptyset, E, s) := \prod_{i=1}^{s} \lambda_i(A[E]),$$

and if F is nonempty and $A[F]$ is invertible, then

$$\begin{aligned}&v(A, F, E, s) \\ &\quad \leq b(A, F, E, s) := \det(A[F]) \prod_{i=1}^{s-f} \lambda_i(B(F, E)),\end{aligned}$$

where

$$B(F, E) = A[E] - A[E, F]A[F]^{-1}A[F, E].$$

We note that if A is a covariance matrix of a set of random variables indexed from N, then $B(F, E)$ is the covariance matrix of the random variables indexed from E, conditioned on the random variables indexed from F.

Proof. Let S_0 be any subset of s elements from E. Without loss of generality, we may take $E = \{1, 2, \ldots, e\}$ and $S_0 = \{1, 2, \ldots, s\}$. Now apply the eigenvalue interlacing property for $s \leq r \leq e - 1$ to the symmetric order-e matrix $A[E]$. A consequence is that

$$\lambda_i(A[S_0]) \leq \lambda_i(A[E]) \quad \text{for } 1 \leq i \leq s.$$

Hence,

$$\prod_{i=1}^{s} \lambda_i(A[S_0]) \leq \prod_{i=1}^{s} \lambda_i(A[E]).$$

The first part of the result follows by noticing that the product on the left is precisely $\det(A[S_0])$.

If F is nonempty and $A[F]$ is invertible, then

$$\begin{pmatrix} A[F] & A[F, E] \\ A[E, F] & A[E] \end{pmatrix} = \begin{pmatrix} A[F] & 0 \\ A[E, F] & I \end{pmatrix} \begin{pmatrix} I & A[F]^{-1}A[F, E] \\ 0 & B(F, E) \end{pmatrix}.$$

Therefore, $\det(A[F \cup E]) = \det(A[F]) \cdot \det(B(F, E))$. All that remains is to show that $v(B(F, E), \emptyset, E, s - f)$ is bounded above by the product of the $s - f$ greatest eigenvalues of the order-e matrix $B(F, E)$. But this follows from the first part of the result.

We note that bounds based on eigenvalues are not new in combinatorial optimization. Eigenvalue-based bounds for other problems have been established using other ideas such as the Hoffman–Wielandt inequality, the Perron-Frobenius theorem, Rayleigh's principle, the Courant-Fisher theorem and Fan's theorem (see Mohar and Poljak 1992).

4. THE ALGORITHM

We employ a branch-and-bound strategy to calculate an optimal solution to our problem. An important ingredient of our algorithm is the bound described in Section 3. Another essential ingredient is a heuristic for the problem to generate an initial candidate solution as well as subsequent candidate solutions. Branch-and-bound methods are well known in the context of linear/integer-programming approaches to linear objective binary programming problems (see Nemhauser and Wolsey 1988). Such methods naturally stumble upon good feasible solutions as linear programming subproblems are solved. Hence, such methods are often quite successful for problems of moderate size even without the use of heuristics to generate good candidate solutions. General purpose heuristics (such as "pivot and complement" (see Balas and Martin 1980) and problem specific heuristics usually improve the situation. Our situation is somewhat different in that the bounding procedure does not naturally discover candidate solutions. The only candidate solutions that are stumbled upon arise from subproblems that only have one feasible solution. Hence, the use of a decent heuristic is vital to the success of our algorithm.

The most natural choices of heuristics are based on both greedy and interchange ideas. We describe a simple version of such a method. Starting with an arbitrary set $S = F(|S| < s)$, we repeatedly replace S with $S \cup i$ and E with $E\backslash i$ provided $|S| < s$ and

$$\det(A[S \cup i]) \geq \det(A[S \cup j]) \quad \text{for all } j \in N\backslash S.$$

Next, we replace S with $S\backslash i \cup j$ and E with $E\backslash j \cup i$, if $\det(A[S\backslash i \cup j]) > \det(A[S])$ for some $i \in S\backslash F$ and $j \in E$. We repeat these interchanges until there are no such i and j. The calculations are simplified by noting that

$$\det(A[S\backslash T \cup j])$$

$$= \det(A[S\backslash T])(A[j] - A[j, S\backslash T]A[S\backslash T]^{-1}A[S\backslash T, j]),$$

which, for each fixed T, is a simple calculation for all j using a Cholesky factorization of $A[S\backslash T]$. In the initial greedy phase we take $T = \emptyset$, and in the interchange phase we take $T = \{i\}$.

An alternative to the greedy initialization is a dual-greedy approach, where we start with $S = N$, and while $|S| > s$, we repeatedly replace S with $S\backslash i$ provided $i \in S\backslash F$ satisfies

$$\det(A[S\backslash i]) \geq \det(A[S\backslash j]) \quad \text{for all } j \in S\backslash F.$$

We initiate the algorithm by running the heuristic to obtain a candidate solution S_c, and initialize the lower bound by setting $LB := \det(A[S_c])$. We also initialize the set $\mathcal{L}$ of active subproblems with the singleton (A, F, E, s), calculate $b(A, F, E, s)$, and initialize the global upper bound by setting $UB := b(A, F, E, s)$.

At a general step of the algorithm, if $UB > LB$, we remove an active subproblem (A, F', E', s) from $\mathcal{L}$ and select some $i \in E'$ as a branching index. We append $(A, F', E'\backslash i, s)$ to $\mathcal{L}$, and calculate $b(A, F', E'\backslash i, s)$, provided that $|F'| + |E'| - 1 > s$. If $|F'| + |E'| - 1 = s$, then the only feasible solution to this new subproblem is $S := F' \cup E'\backslash i$. In this latter case, if $\det(A[S]) > LB$, we replace S_c with S and set $LB := \det(A[S])$. We also append $(A, F' \cup i, E'\backslash i, s)$ to $\mathcal{L}$, and calculate $b(A, F' \cup i, E'\backslash i, s)$, provided that $|F'| + 1 < s$. If $|F'| + 1 = s$, then the only feasible solution to this subproblem is $S := F' \cup i$; in this case, we replace S_c with S and set $LB := \det(A[S])$ if $\det(A[S]) > LB$. We update the global upper bound by setting $UB := \max_{L \in \mathcal{L}} b(L)$. We may attempt to increase the lower bound at any step by rerunning the heuristic based on information obtained from the current state of $\mathcal{L}$. This is prudent whenever the size of $\mathcal{L}$ becomes intolerably large. The algorithm is finite and terminates with S_c optimal when $LB = UB$. We can terminate early if we are willing to be satisfied when LB is close enough to UB.

Finally, we remark that it is straightforward to modify our algorithm to solve $\mathbf{P}(\min, \|)$ and $\mathbf{P}(\max, \cdot)$.

5. COMPUTATION

The algorithm was implemented in FORTRAN 77 and run on a lightly loaded HP 9000/730 workstation (using the F77 compiler with the -K and -O options, under HP-UX). Eigenvalue calculations and other matrix manipulations were performed by calls to appropriate EISPACK and LINPACK subroutines and functions (see Smith et al. 1970, and Dongarra et al. 1978) obtained from the NETLIB. To find a good feasible solution we initialized the interchange heuristic with the outputs of both the greedy heuristic and the dual-greedy heuristic, and chose the best solution thus obtained. After bound calculation, subproblems not fathomed by bounds were placed on a FIFO queue. After any unfathomed subproblems were placed on the queue, the global upper bound was recalculated (if the bound of the parent subproblem was equal to the global upper bound) by making a pass through the queue. Upon fathoming a subproblem by feasibility, the lower bound was updated; rather than making an additional pass through the queue to delete active subproblems that become fathomed by bounds, we waited until such subproblems rose to the top of the queue. If storage problems arise, this is an obvious place where one can sacrifice a bit of time to economize on memory. One could also keep the subproblems in a heap ordered by bound; this would make it easier to update the global

Table I
Network Expansion Problems: Ohio Valley

Problem Number: $n/n - f/s - f$	Initial Absolute Entropy Gap	UB Calls	Maximum Number Active Subproblems	Wall Seconds: Greedy/Swap/ UB/Total
1: 52/16/8	0.18149914	31	1	0/0/1/2
2: 63/27/13	0.56583546	323	15	1/0/6/7

upper bound and facilitate a strategy of branching to greedily decrease the upper bound. The only information that we stored for the subproblems were the sets of forced and candidate indices. We did not attempt to store matrix factorization or eigenvector information that might have been used to speed bound calculation for subsequent subproblems. Furthermore, we did not investigate "sophisticated" branching strategies (based on eigenvector information, for example). We simply branched on the least eligible-index. In short, we kept it simple.

We ran our code on Guttorp et al.'s problems of selecting 8 of 16 potential sites (problem 1) and 13 of 27 potential sites (problem 2) to add to a current 36-station network. In both cases the greedy solution without any pairwise interchanges proved to be optimal. In both cases, our program verified optimality quite quickly. Results are summarized in Table I. Each entry in the column labeled Initial Absolute Entropy Gap is the difference between the logarithm of the initial upper bound and the entropy of the solution generated by the heuristic; we use an *absolute* measure since it is invariant under matrix scaling. These initial gaps give some a priori measure of problem difficulty. The column labeled UB Calls contains the numbers of times that the upper bound of Proposition 1 was calculated. The column labeled Max Number Active Subproblems records the maximum length of the queue of subproblems to be solved during the course of the algorithm. The final column tabulates the time spent in different parts of the algorithm.

We also ran our code on ten random problems (problems 3–12) to get some idea of the robustness of our approach. Each input matrix A arises by generating n points (x_i, y_i) uniformly in the unit square. The first f points are required to be in the solution, and we want to choose $s - f$ more, out of the remaining $n - f$ points. The matrix A is defined by setting $A[i, j] := (2 - d(i, j))/10$ for $i \neq j$, where $d(i, j)$ is the square of the Euclidean distance from (x_i, y_i) to (x_j, y_j), and setting $A[i, i] := 1/n + \sum_{j:j\neq i} A[i, j]$. Off-diagonal entries reflect covariances that decrease with the distance between the point pairs. The matrix is defined so that it is diagonally dominant, hence it is positive definite. Problems 3–7 have the same dimensions as problem 2. The dimensions of problems 8–12 make them appear to be more difficult. The results are summarized in Table II. In all cases, the greedy solution without any pairwise interchanges proved to be optimal. Solution statistics for problems 3–7 indicate that problems from our random distribution appear to be modestly more difficult than real problems of comparable size (despite the smaller initial gaps than problem 2). One would guess that problems 8–12 might be a bit more difficult than real problems of comparable size. We note that although the running times of the code on problems 8–12 are significantly greater than on problems 1–7, the number of feasible solutions of the problems 8–12 is significantly larger.

Finally, we obtained another covariance matrix from N. D. Le and J. V. Zidek. The covariance matrix is estimated from sulfate concentration records recorded at a set of 36 stations, after the data had been appropriately deseasonalized and detrended. For $s = 26, 27, \ldots, 35$, we solved the problem with $f = 0$ (problems 13–22). Results are summarized in Table III. These problems were considerably more difficult than those discussed

Table II
Network Expansion Problems: Random

Problem Number: $n/n - f/s - f$	Initial Absolute Entropy Gap	UB Calls	Maximum Number Active Subproblems	Wall Seconds: Greedy/Swap/ UB/Total
3: 63/27/13	0.09034067	505	18	1/0/10/12
4: 63/27/13	0.08059517	245	9	0/1/5/7
5: 63/27/13	0.09713827	455	22	0/1/10/11
6: 63/27/13	0.08854448	337	21	1/0/7/8
7: 63/27/13	0.10793617	933	45	0/1/20/21
8: 75/50/25	0.17567235	15,647	524	1/2/394/397
9: 75/50/25	0.18478354	8,725	261	1/2/202/205
10: 75/50/25	0.14474644	2,619	96	1/2/68/71
11: 75/50/25	0.16473960	4,829	193	2/1/128/131
12: 75/50/25	0.16728875	4,869	159	2/1/134/137

Table III
Network Design Problem: Sulphate Monitoring

Problem Number: $n/n - f/s - f$	Initial Absolute Entropy Gap	UB Calls	Maximum Number Active Subproblems	Wall Seconds: Greedy/Swap/ UB/Total
13: 36/36/26	8.27061608	1,032,573	67,338	0/1/4478/4576
14: 36/36/27	7.88852013	621,037	38,278	0/1/2809/2851
15: 36/36/28	7.41078815	315,446	18,559	0/1/1482/1509
16: 36/36/29	6.83095104	128,547	7,247	1/0/639/646
17: 36/36/30	6.27408022	51,675	2,630	0/1/269/273
18: 36/36/31	5.50751516	19,101	875	0/0/105/106
19: 36/36/32	4.64422768	6,772	283	0/1/37/40
20: 36/36/33	3.71884912	2,331	81	1/0/14/15
21: 36/36/34	2.44959923	571	17	0/0/3/4
22: 36/36/35	1.29342509	71	2	0/1/1/2

above. We note that in all problems, the dual-greedy heuristic discovered the solution that turned out to be optimal (the greedy heuristic discovered the optimal solution only for $s = 33, 34, 35$, although the greedy heuristic followed by swapping was also successful for $s = 30, 31, 32$).

In all of our experiments, the time spent by the heuristic is negligible.

Our experimental evidence indicates that the bound provided by Proposition 1 is quite good. In all cases the number of subproblems explored is always an extremely small fraction of the number of the number of feasible solutions $\binom{n-f}{s-f}$ (note that the number of feasible solutions is certainly a lower bound on the number of subproblems that one would have to solve if we never fathomed using the upper bound of Proposition 1). In addition, the maximum number of active subproblems is always even smaller. In the extreme, for problem 1, for every pair of subproblems created, the bound enabled us to fathom one of the subproblems; this is the best possible situation and points to the quality of the bound.

6. REMARKS

Our algorithm is quite effective on the data sets that we applied it to. It would be interesting to see how robust our approach is across input matrices from different application areas.

In a subsequent paper (Ko, Lee and Wayne 1994), we study the related statistical design problem of chosing a prespecified number of design rows from the design matrix of a linear model, so as to minimize the generalized variance of the least-squares parameter estimators (see Mitchell 1974a, b). For that problem, we present complexity results, a bound based upon singular values, as well as computational results using the bound in a branch-and-bound algorithm.

ACKNOWLEDGMENT

The authors thank Professor J. V. Zidek (University of British Columbia) for bringing this problem to their attention and for providing references on the design of spatial sampling networks and their environmental applications. They also thank him and Mr. N. D. Le for providing the data sets used in this study. They thank Professor A. K. Kelmans (Rutgers University) for pointing out the reference by Kelmans and Kimelfeld.

The research of the first author was supported by the Centrum voor Wiskunde en Informatica, Stichting Mathematisch Centrum, Amsterdam, The Netherlands, and a Postdoctoral Fellowship from the Natural Sciences and Engineering Research Council (NSERC) of Canada. The research of the second author was conducted in part at CORE, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium, and was supported in part by a CORE Fellowship and a Yale Senior Faculty Fellowship. The research of the third author was supported by a grant from NSERC.

REFERENCES

E. BALAS, AND R. MARTIN. 1980. Pivot and Complement: A Heuristic for 0-1 Programming. *Mgmt. Sci.* **26,** 86–96.

P. J. DAVIS. 1979. *Circulant Matrices*. John Wiley, New York.

Y. DODGE, V. V. FEDOROV AND H. P. WYNN. 1988. *Optimal Design and Analysis of Experiments*. Elsevier.

J. DONGARRA, J. R. BUNCH, C. B. MOLER AND G. W. STEWART. 1978. *LINPACK Users Guide*. SIAM Publications, Philadelphia.

J. EDMONDS. 1967. Systems of Distinct Representatives and Linear Algebra. *J. Res. Natl. Bur. Stand.* **71B,** 241–245.

V. V. FEDOROV, AND P. HACKL. 1990. Optimal Experimental Design: Spatial Sampling. Institut fur Statistik, Wirtschaftsuniversitat, Vienna, Austria.

M. R. GAREY, AND D. S. JOHNSON. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco.

M. GROTSCHEL, L. LOVÁSZ AND A. SCHRIJVER. 1988. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, New York.

P. GUTTORP, N. D. LE, P. D. SAMPSON AND J. V. ZIDEK. 1992. Using Entropy in the Redesign of an Environmental Monitoring Network. Technical Report #116, Department of Statistics, The University of British Columbia.

A. K. KELMANS, AND B. N. KIMELFELD. 1983. Multiplicative Submodularity of a Matrix's Principal Minor as a Function of the Set of its Rows, and Some Combinatorial Applications. *Discr. Math.* **44,** 113–116.

C. W. KO, J. LEE AND K. WAYNE. 1994. A Spectral Bound for *D*-Optimality. Manuscript.

T. J. MITCHELL. 1974a. An Algorithm for the Construction of 'D_k-Optimal' Experimental Designs. *Technometrics* **16,** 203–210.

T. J. MITCHELL. 1974b. Computer Construction of 'D_k-Optimal' First-Order Designs. *Technometrics* **16,** 211–220.

B. MOHAR, AND S. POLJAK. 1992. Eigenvalues in Combinatorial Optimization. Preprint Series, Vol. 30, #371, Department of Mathematics, University of Ljubljana.

G. L. NEMHAUSER, AND L. A. WOLSEY. 1981. Maximizing Submodular Set Functions: Formulations and Algorithms. In *Studies on Graphs and Discrete Programming*, P. Hansen (ed.). North-Holland, Amsterdam, 279–301.

G. L. NEMHAUSER, AND L. A. WOLSEY. 1988. *Integer and Combinatorial Optimization*. John Wiley, New York.

M. C. SHEWRY, AND H. P. WYNN. 1987. Maximum Entropy Sampling. *J. Appl. Statist.* **46,** 165–170.

B. T. SMITH, J. M. BOYLE, Y. IKEBE, V. C. KLEMA AND C. B. MOLER. 1970. Matrix Eigensystem Routines: EISPACK Guide, Second Edition. Springer-Verlag, New York.

J. H. WILKINSON. 1965. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, England.

S. WU, AND J. V. ZIDEK. 1992. An Entropy Based Review of Selected NADP/NTN Network Sites for 1983–1986. *Atmos. Environ.* (to appear).