Mathematical Programming manuscript No.
(will be inserted by the editor)

Samuel Burer · Jon Lee

Solving Maximum-Entropy Sampling Problems Using Factored Masks

February 28, 2005

Abstract. We present a practical approach to Anstreicher and Lee's masked spectral bound for maximum-entropy sampling, and we describe favorable results that we have obtained with a Branch-&-Bound algorithm based on our approach. By representing masks in factored form, we are able to easily satisfy a semidefiniteness constraint. Moreover, this representation allows us to restrict the rank of the mask as a means for attempting to practically incorporate second-order information.

Introduction

Let n be a natural number, and let $N := \{1, 2, \ldots, n\}$. Let C be an order-n symmetric positive definite matrix. Let s be an integer between 0 and n. For subsets S and T of N, let $C[S, T]$ denote the submatrix of C having rows indexed by S and columns indexed by T. The *maximum-entropy sampling problem* is to calculate

$$z(C, s) := \max \{\ln \det C[S, S] \ : \ S \subset N, \ |S| = s\}.$$

This fundamental problem in the *design of experiments* was introduced in [18] and first used in a monitoring design context in [8] (also see [9,16,17,20]).

From the perspective of algorithmic optimization, the problem has been studied extensively; see [1–3,10–12,15] and the surveys [13,14]. Exact algorithms are based on the Branch-&-Bound framework. Crucial to such an approach is a good upper bound on $z(C, s)$.

In §1 we describe the masked spectral bound of [3]. In §2, we describe two first-order rank-restricted algorithms for quickly obtaining a good mask. In §3, we compare our methods with one another and with the affine-scaling algorithm of [3]. We also examine the issue of how to choose the rank. In §4, we describe our Branch-&-Bound implementation and results. Finally, in §5, we mention possibilities for incorporating second-order information, and in §6, we discuss some future directions.

Samuel Burer: Department of Management Sciences, University of Iowa, e-mail: samuel-burer@uiowa.edu. Supported in part by NSF Grant CCR-0203426.

Jon Lee: Department of Mathematical Sciences, T.J. Watson Research Center, IBM, e-mail: jonlee@us.ibm.com.

1. The masked spectral bound

Anstreicher and Lee introduced the masked spectral (upper) bound (see [3]) for $z(C, s)$. A *mask* is any symmetric, positive semidefinite matrix having ones on the diagonal, i.e., any $X \succeq 0$ having $\mathrm{diag}(X) = e$. We define the associated *masked spectral bound* as

$$\xi_{C,s}(X) := \sum_{l=1}^{s} \ln\left(\lambda_l\left(C \circ X\right)\right) \ ,$$

where $\circ$ represents the Hadamard product of matrices and eigenvalues $\lambda_1, \ldots, \lambda_n$ are in nonincreasing order. Validity of the masked spectral bound is based on: (i) *Oppenheim's inequality*, i.e., $\det A \leq \det A \circ B / \prod_{j=1}^{n} B_{jj}$, where $A \succeq 0$ and $B \succeq 0$; and (ii) the eigenvalue inequalities $\lambda_l(A) \geq \lambda_l(B)$, where $A \succeq 0$, and B is a principal submatrix of A.

The masked spectral bound is a generalization of the *spectral partition bound* of [10] (take X to be block diagonal, with blocks of 1's), which itself is a generalization of both the *eigenvalue bound* of [11] (take $X = ee'$) and the *diagonal bound* of [10] (take $X = I$). The spectral partition bound can produce much better bounds than the eigenvalue and diagonal bounds, but there is no known practical methodology for efficiently choosing a near-optimal partition of N, which describes the block structure of the mask (see [10]). Some success has been reported using the following method for calculating the so-called "one-big partition" of N (see [10] and [3]): (i) let S be a heuristic solution of the maximum-entropy sampling problem; (ii) the associated *one-big partition* of N has one block S and the elements of $N \setminus S$ as singletons.

The motivation for working with the masked spectral bound is to try to use methods of continuous optimization to get the strength of the combinatorial bounds (i.e., the spectral partition bounds and its specializations) but more efficiently. Specifically, we try to find a mask yielding a good bound. Finding an optimal mask is a problem of minimizing a nondifferentiable nonconvex function subject to a semidefiniteness constraint. That sounds daunting, except for the following inter-related points: (i) for our purposes, we do not need a global minimum; (ii) in our experience the nondifferentiability is not so serious; (iii) in our experience, local solutions obtained from different starting points are not wildly different in value; and (iv) if we do badly with the upperbound for a subproblem, we can make up for it with the children.

An important point is that function, gradient and Hessian evaluations for ξ are expensive. All of these are based on a single (expensive) eigensystem calculation. Function evaluations require eigenvalues, gradients require the eigenvectors as well, and Hessians further require some non-negligible arithmetic. Considering that we are embedding these calculations in Branch-&-Bound, with the goal for each subproblem of trying to push its upper bound below the value of a global lower bound, we can not afford to spend much time minimizing; so we need to find a way to descend quickly in just a few steps.

In [3], Anstreicher and Lee proposed a method for minimizing ξ which was based on ideas coming from the affine scaling algorithm for linear programming.

They considered different algorithmic variants, such as short- and long-step approaches, and they demonstrated the ability to achieve good bounds (relative to the one-big, eigenvalue, and diagonal bounds), even in the face of the nonconvexity and nondifferentiability of ξ. As their main goal was to assess the quality of the optimized masked spectral bound, they stopped short of evaluating the bound in a Branch-&-Bound algorithm.

2. Rank-restricted masks

Our idea is to work with rank-restricted masks in factored form. That is, we consider masks of the form $X := VV'$, where V is $n \times k$ and $1 \leq k \leq n$. As we try to find a mask yielding a low bound ξ, we can hope to make more rapid progress owing to two factors: (i) the semidefiniteness restriction is handled implicitly by the factored form; and (ii) by choosing smaller k, we work with many fewer variables (kn rather than $n(n-1)/2$), which may lead to faster convergence of a steepest-descent type method and also opens up second-order methods as a realistic possibility. As it turns out in our experiments, (i) proves to be more beneficial than (ii).

We note that the restriction to low-rank PSD matrices has been used successfully in other contexts [4–6]. In fact, the first algorithm that we describe borrows its key ideas from [4].

Our usage of rank-restricted masks will be based on three functions. The first normalizes the rows of V, i.e.,

$$h(V) := \operatorname{Diag}\left(\operatorname{diag}\left(VV'\right)^{-1/2}\right) V;$$

the second takes a normalized $\bar{V}$ and calculates the corresponding masked spectral bound, i.e.,

$$g(\bar{V}) := \xi_{C,s}(\bar{V}\bar{V}');$$

and the third is the composite function $f := g \circ h$. Note that h is only well-defined if all rows of V are nonzero.

We propose a steepest-descent type algorithm for minimizing $f(V)$, and for this, we consider the gradient of f. Let $u_l(C \circ \bar{V}\bar{V}')$ be an eigenvector, of unit Euclidean norm, associated with $\lambda_l(C \circ \bar{V}\bar{V}')$. Let U be the $n \times s$ matrix having columns u_l ($l = 1, \ldots, s$), and let Σ be the order-s square diagonal matrix of λ_l. Then, as long as $\lambda_s > \lambda_{s+1}$, we have that the gradient of f at V is the matrix

$$\nabla f(V) = \operatorname{Diag}(d)\, \nabla g(\bar{V}) - \operatorname{Diag}\left(\operatorname{diag}\left(\nabla g(\bar{V})V'\right) \circ d^3\right) V;$$

where

$$\begin{aligned} d &= \operatorname{diag}\left(VV'\right)^{-1/2}, \\ \bar{V} &= h(V), \\ \nabla g(\bar{V}) &= 2\left(C \circ U\Sigma^{-1}U'\right)\bar{V}. \end{aligned}$$

This can be derived using standard results concerning symmetric functions of eigenvalues (see [19], for example). Note that we must define u_l properly when $\lambda_l = \lambda_{l+1}$ for any $l = 1, \ldots, s-1$; in such situations, we just take care that the associated $\{u_l\}$ form an orthonormal basis for each of the eigenspaces corresponding to distinct eigenvalues.

Our steepest-descent algorithm (Algorithm 1) is described in Figure 1. Due to

0. Initialize V, $t_{\max}$, $k_{\max}$ and $0 < \beta < 1$. Set $t = 0$.
1. Let $k := 0$ and $t := t + 1$. If $t > t_{\max}$ then STOP.
2. Let $W := V - \beta^k \nabla f(V)$.
3. If $f(WW') < f(VV')$ or $k = k_{\max}$, then let $V := W$ and GOTO 1. Otherwise, let $k := k+1$ and GOTO 2.

Fig. 1. Basic Descent Algorithm

the possible non-differentiability of f, an important ingredient of the algorithm is the acceptance of a non-improving step if descent is not detected after a certain number ($k_{\max}$) of trials. Values such as $k_{\max} := 2$ and $\beta := 1/3$ are practical. In what follows, we will describe the results of our experiments with various values for $t_{\max}$ and various choices of the initial V. For example, in the Branch-&-Bound results of §4, we will take $t_{\max} \leq 5$.

Anstreicher and Lee [3] used a similar back-tracking approach for their affine-scaling algorithm. In particular, they also used parameters $k_{\max} = 2$ and $\beta = 1/3$.

An alternative to Algorithm 1 that we experimented with (referred to as Algorithm 2) is given in Figure 2. Effectively, the main difference between the

0. Initialize $\bar{V}$ with unit-length rows, $t_{\max}$, $k_{\max}$ and $0 < \beta < 1$. Set $t = 0$.
1. Let $k := 0$ and $t := t + 1$. If $t > t_{\max}$ then STOP.
2. Let $\bar{W} := h(\bar{V} - \beta^k \nabla g(\bar{V}))$.
3. If $g(\bar{W}\bar{W}') < g(\bar{V}\bar{V}')$ or $k = k_{\max}$, then let $\bar{V} := \bar{W}$ and GOTO 1. Otherwise, let $k := k+1$ and GOTO 2.

Fig. 2. Alternative Descent Algorithm

two algorithms is how the normalization h is handled. In Algorithm 1, the normalization is handled implicitly by minimizing the composite function f, while in Algorithm 2, it is enforced manually as part of the minimization of g.

Another interpretation of Algorithm 2 is that the direction $-\nabla g(\bar{V})$ defines a path along the feasible set of normalized matrices. Even if g is differentiable at $\bar{V}$, however, this search path is not necessarily a path of descent, though it is nearly always in practice.

3. Experiments with rank-restricted masks

In this section, we detail our experience with Algorithms 1 and 2. A few preliminary remarks are in order. First, the data for our experiments, which has $n = 63$ and was also used in [3], comes from an environmental monitoring application (see [9]).

Second, we will often compare bounds on the *original* and *complementary* problems. This terminology refers to the fact that, using the identity

$$\ln \det C[S, S] = \ln \det C + \ln \det C^{-1}[N \setminus S, N \setminus S],$$

any bound for the complementary problem of choosing a maximum entropy set of $n - s$ points with respect to the covariance matrix C^{-1} translates to a bound for the original problem (see [1,2]). In practice, the same class of bound (e.g., the one-big bound) yields very different values when calculated with respect to the original and complementary problems.

Finally, a primary basis for comparison will be the gap between a calculated bound and the entropy value of a heuristic solution, which is obtained by a greedy algorithm.

3.1. Comparison of Algorithms 1 and 2

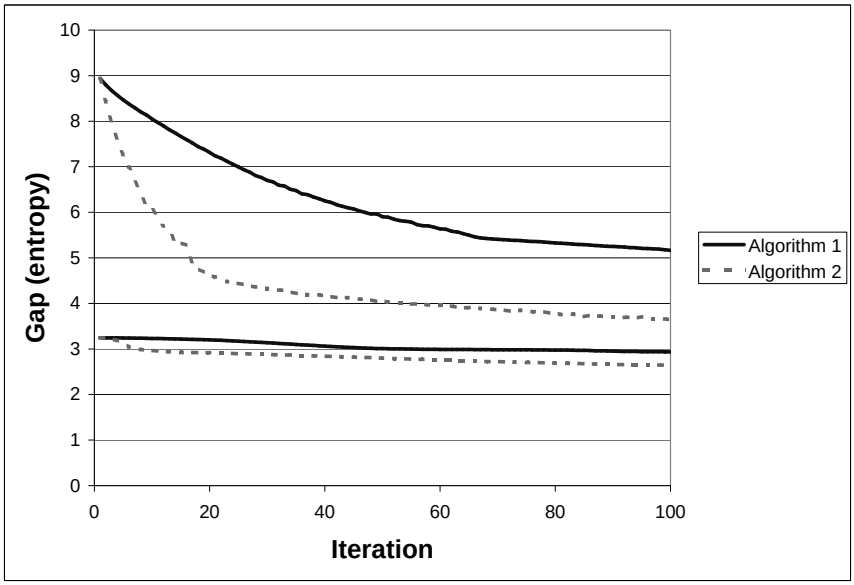
We first compare Algorithm 1 with Algorithm 2. Because both algorithms require roughly the same amount of work per iteration, we have found that the running times of the algorithms do not differ significantly. Accordingly, we are primarily interested in comparing convergence properties.

Figure 3 depicts a typical outcome when comparing Algorithms 1 and 2 on different instances of the maximum-entropy sampling problem with different values of k. For this specific experiment, we ran both algorithms from the same, random starting point for 100 iterations with $(n, s, k) = (63, 31, 63)$ on both the original and complementary problems. The curves indicate convergence by depicting entropy gap versus iteration number.

As Figure 3 demonstrates, Algorithm 2 converges more quickly than Algorithm 1 and also achieves better gaps. This is intriguing behavior in light of the fact that Algorithm 1 employs true descent directions for f, while Algorithm 2 uses a kind of heuristic search path. One hypothesis as to why we see this behavior with Algorithm 2 as follows: its search path may be perturbing us away from points where ξ is not differentiable (e.g., Lee and Overton (private communication) have observed nondifferentiability at true local minimizers obtained by the gradient-sampling algorithm (see [7])).

Another comparison of Algorithms 1 and 2, which supports a similar conclusion as Figure 3, will be given in §3.3.1. Nonetheless, we will advocate the use of Algorithm 1 in §4. Specific reasons for doing so will be explained in that section.

Fig. 3. Comparison of the gaps produced by Algorithm 2 and Algorithm 1 when each is run for *100* iterations from the *same, random* starting point on both the *original and complentary* problems. Relevant parameters are $(n, s, k) = (63, 31, 63)$. The top two curves represent the original problem; the bottom two curves represent the complementary problem.



3.2. Comparison with affine scaling

We next compare Algorithm 1 with the affine scaling algorithm of Anstreicher and Lee [3], which we refer to as Algorithm AS. In order to obtain as fair a comparison as possible, we obtained their code, which was written in Matlab, and constructed our own Matlab code for Algorithm 1. All tests for both algorithms were initialized with the same starting point as described in [3]. In particular, since Algorithm AS is an interior-point method, we took $k = n$ for Algorithm 1 in all comparisons between the two methods.

Our comparisons are based on the following three criteria, each of which has two possible realizations (for an overall total of eight comparisons):

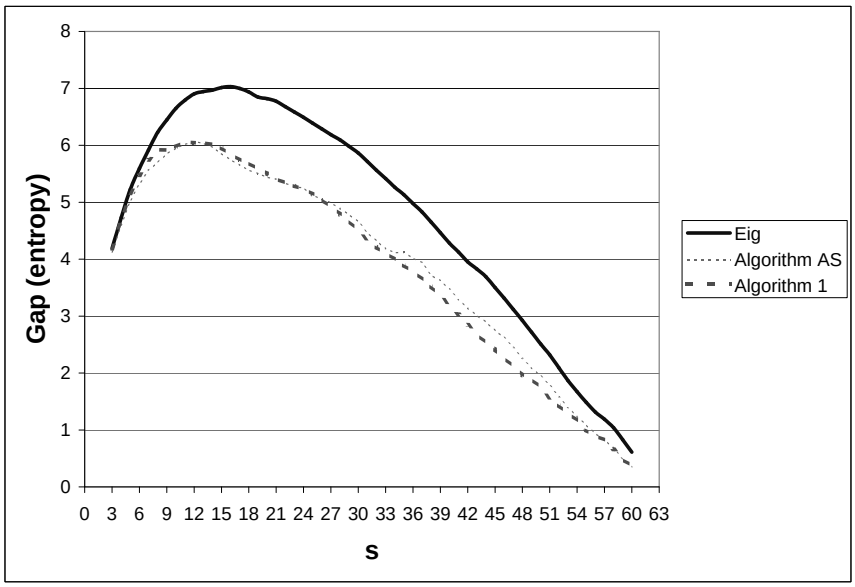
(a) *initial* V — a full-rank perturbation of either the eigenvalue or one-big mask;
(b) $t_{\max}$ — either 5 or 100;
(c) *problem type* — either original or complementary problem.

In each of the eight comparisons, we ran the algorithms for all s between 3 and 60, inclusive.

The purpose of comparing on the basis of $t_{\max}$ is to observe performance early and late in the algorithm. Early performance (e.g., $t_{\max} = 5$) gives an indication of how the algorithms would perform in a Branch-&-Bound algorithm, where we can only afford to do a few steps to obtain a bound at each node in the tree. On the other hand, late performance (e.g., $t_{\max} = 100$) gives an indication of the overall performance and robustness of the algorithms.

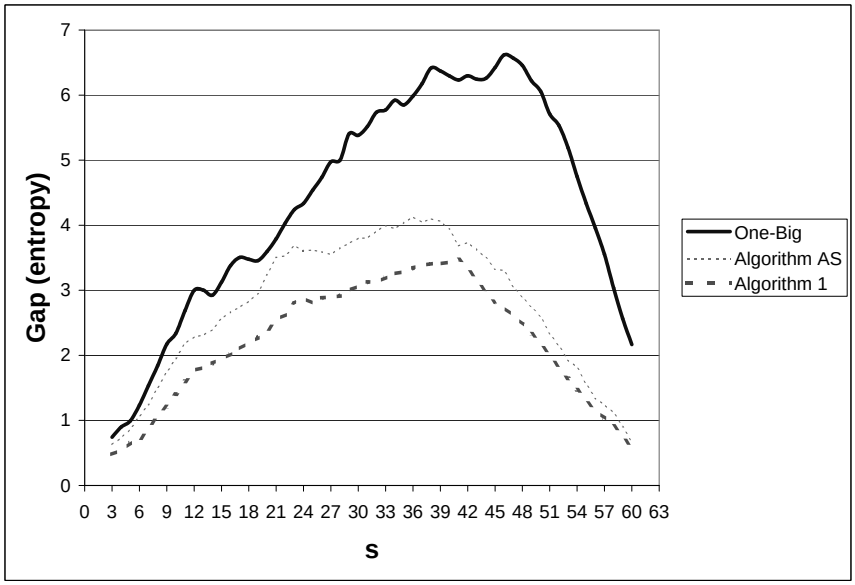
Over all eight comparisons, we found similar behavior, namely that Algorithm 1 did at least as well as — and often significantly better than — Algorithm AS. Accordingly, in the interest of space, we provide two representative comparisons of the eight total in Figures 4 and 5.

Fig. 4. Comparison of the gaps produced by Algorithm 1 and Algorithm AS after *5* steps when initialized near the *eigenvalue* mask on the *complementary* problem, for $s = 3, \ldots, 60$. The gap given by the eigenvalue mask is also shown for reference.



Another important criterion for comparison is running time. For both Algorithm AS and Algorithm 1, the work per iteration is dominated by the eigenvalue and eigenvector calculations associated with the function ξ. Recall also that both algorithms use the same line search back-tracking technique. As such, the running times for the two methods (for a fixed number of iterations) are quite comparable and so are not a source of differentiation.

Fig. 5. Comparison of the gaps produced by Algorithm 1 and Algorithm AS after *100* steps when initialized near the *one-big* mask on the *original* problem, for $s = 3, \ldots, 60$. The gap given by the spectral partition mask is also shown for reference.



Overall, our experiments lead us to favor Algorithm 1 over Algorithm AS. In particular, we will use Algorithm 1 exclusively for our Branch-&-Bound experiments in §4.

3.3. Choosing the rank

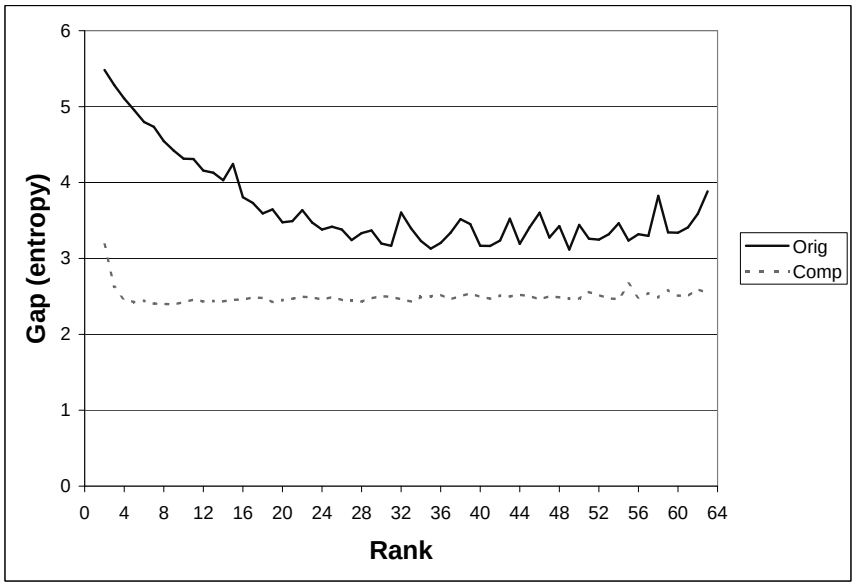
An important parametrization issue concerns how the rank k affects the masked spectral bound. In particular, if we can achieve nearly the same bound by running Algorithm 1 with a smaller k, we could hope that this would lead to decreased computational requirements. Of course, practically speaking, we need to know precisely which rank to choose and how much computational savings can be expected.

Unfortunately, we have found it difficult to provide any guidelines on which rank to choose. Moreover, the computational savings that we have been able to achieve from reducing the rank are insignificant due to the fact that the eigensystems calculations associated with the function ξ dominate the running of Algorithm 1. We illustrate these points by means of some example runs.

For each k between 2 and 63, we ran Algorithm 1 twice with $s = 31$, once on the original problem and once on the complementary problem. For both runs, we initialized V randomly and took $t_{\max} = 1000$, that is, we did 1000 iterations. The purpose of so many iterations was to give Algorithm 1 sufficient opportunity

to reduce the bound as far as possible. As before, our basis of comparison was the entropy gap. Figure 6 depicts the results.

Fig. 6. Gaps produced by Algorithm 1 after *1000* steps when initialized *randomly* on the *original* and *complentary* problems, for ranks $k = 2, \ldots, 63$.



From Figure 6, it appears that the optimal rank (that is, the smallest rank that yields the best possible bound) for the original problem is approximately $k = 30$. On the other hand, for the complementary problem, the optimal rank appears to be around $k = 5$. This example illustrates the difficulty in choosing the optimal rank *a priori*.

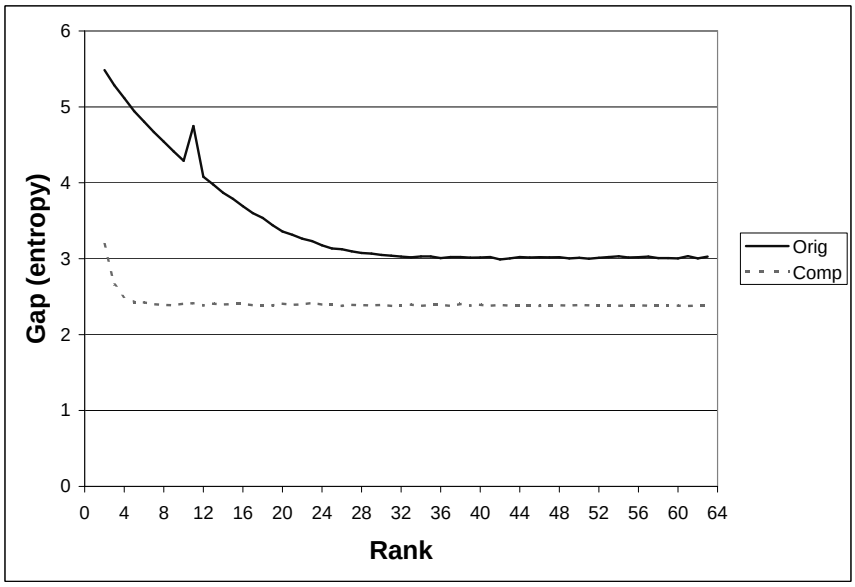
For the original problem, the average running time over all $k = 2, \ldots, 63$ was 10.8 seconds with a standard deviation of 0.3 seconds; for the complementary problem, the numbers were 24.7 and 1.4, respectively. So, the timings showed little variability with respect to the rank k. As mentioned above, this is due to the dominance of the eigensystem calculations.

Overall, a conclusion of our experiments is that, when running Algorithm 1, it is reasonable to take $k = n$ since the resulting running time will not be significantly more than with smaller k. On the other hand, our experiments support the choice of lower k to get comparable bounds — if a way can be found to exploit the lower rank for better convergence and/or running times.

3.3.1. Another comparison of Algorithms 1 and 2 The above experiments give us an additional opportunity to compare Algorithm 1 with Algorithm 2. In Figure 7, we replicate the experiments of Figure 6 except this time with Algorithm

2. Notice that, overall, Algorithm 2 converges more reliably than Algorithm 1 for

Fig. 7. Same as in Figure 6, except that Algorithm 2 is run instead of Algorithm 1.



varying ranks. It is also clear that Algorithm 2 achieves slightly better gaps. This evidence provides additional support for the earlier conclusion that Algorithm 2 is more robust than Algorithm 1.

4. Incorporating in Branch-&-Bound

The Branch-&-Bound approach to maximum-entropy sampling was first described in [11]. Branching is done on single elements of N — either forcing a single element into the set S or barring a single element from S. Thus, at any node in the Branch-&-Bound tree, the corresponding subproblem is determined by forcing a particular set F of f elements into S and barring a particular set U of u elements from S. It remains then to optimize

$$\max \{\ln \det C[S,S] \ : \ S \subset N \setminus U,\ F \subset S,\ |S| = s\}\,.$$

By the Schur complement theorem, this problem is in turn equivalent to choosing a set T of $s-f$ elements from the set $N \setminus (U \cup F)$, so as to maximize the conditional entropy $\ln \det C_{F,U}[T,T]$ (plus the constant $\ln \det C[F,F]$), where

$$C_{F,U} := C[N \setminus F \setminus U, N \setminus F \setminus U] - C[N \setminus F \setminus U, F](C[F,F])^{-1}C[F, N \setminus F \setminus U].$$

In other words, the task at each node is to determine $z(C_{F,U}, s-f)+\ln\det C[F,F]$, which is itself an instance of the maximum-entropy sampling problem. Hence, any bound developed for the maximum-entropy sampling problem may be used throughout the Branch-&-Bound tree.

We adapted the Branch-&-Bound implementation of [1,2], which was written in the C programming language. We kept all default options, including the decision rules for selecting the next node in the tree for branching (the node with the largest upper bound) and for selecting the specific index to branch on (the largest index not already in F or U). One enhancement that we did make, however, was the calculation of an initial global lower bound via a heuristic solution of the maximum-entropy sampling problem, which was obtained by a greedy routine.

Our goal was to determine whether optimizing the masked spectral bound at each node of the tree is a useful bounding strategy. We compare with the following bounding strategy (for simplicity, *orig* and *comp* refer to bounds coming from the original and complementary problems, respectively):

> *Fixed bounding strategy.* For a fixed type of masked spectral bound (i.e., eigenvalue, diagonal, or one-big), compute *orig* and/or *comp* according to the following steps:
> 1. If *orig* < *comp* for the parent, then calculate *orig*; otherwise, *comp*.
> 2. If the calculated bound does not fathom, then also calculate the remaining bound.

Note that, for the eigenvalue bound, *orig* = *comp*, so the above steps simplify.

One consequence of the default bounding strategy is that, high in the tree where fathoming is less likely to occur, it is likely that both *orig* and *comp* will be unnecessarily calculated at each node. We found this drawback to be outweighed by the benefit of incorporating both *orig* and *comp*. In particular, the single-minded strategies of just computing either *orig* or *comp* throughout the tree did not work as well.

In developing our strategy for optimizing the masked spectral bound, we felt that calculating bounds for both the original and complementary problems at each node would be too expensive. On the other hand, we knew it would be highly beneficial to compute the better of the two. After some experimentation, we arrived at a compromise that was based on the following: we often noticed that different subproblems with the same number of elements fixed in and out of S (i.e., the same f and u) behaved similarly in terms of which bound, *orig* or *comp*, was stronger. So throughout the Branch-&-Bound calculations, we kept track of all pairs (f,u). The first time that a subproblem with a particular (f,u) was encountered, we calculated both *orig* and *comp* and took note of which was stronger. Afterwards, every time the same (f,u) was observed, we would calculate *orig* or *comp* according to our first experience.

Our overall strategy for optimizing the masked spectral bound at each node is as follows:

Optimization bounding strategy. Determine whether to compute *orig* and/or *comp* (as described above). Then, for each bound to be calculated, do the following:

1. Run Algorithm 1 with $t_{\max} = 2$, terminating immediately if fathoming occurs.
2. Using the points generated by Algorithm 1 so far (three points, including the starting point), use quadratic interpolation to judge whether fathoming will occur in the next three iterations.
3. If so, then set $t_{\max} = 5$ and continue Algorithm 1, terminating immediately if fathoming occurs; otherwise, stop.

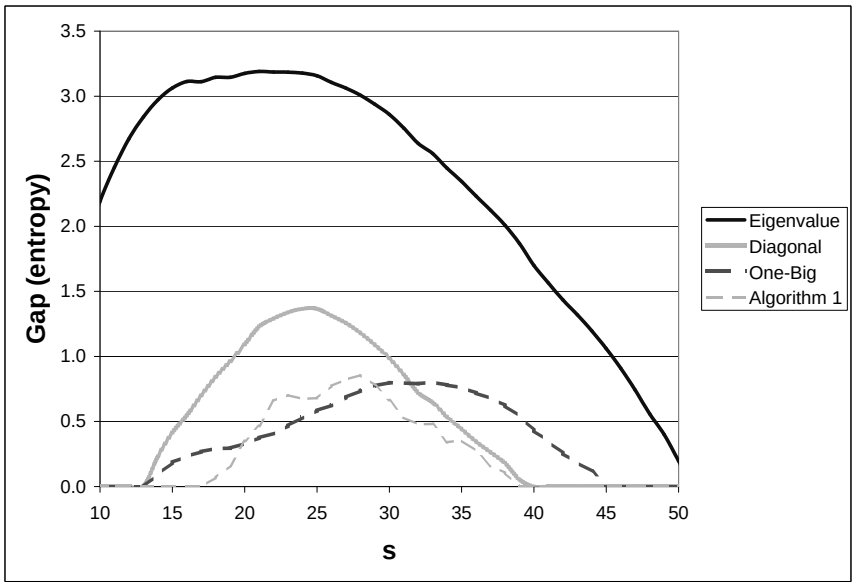
A couple of other details concerning the optimization bounding strategy are worth mentioning. First, in accordance with our discussion in §3, we take full rank, i.e., $k = s - f$, in Algorithm 1 at each node of the tree. Second, instead of initializing V at each node randomly, we attempted to simulate the warm starting of a child node by its parent. (We could not implement a true warm-start strategy due to computer memory limitations.) We did this by keeping a single, shared-memory copy of V — having size $n \times n$ and stored in column-major format — available to all nodes in the tree. A subproblem of size $s-f$ was initialized from the first $(s-f)^2$ entries of V and its final iterate was stored back into the same entries of V. Although clearly heuristic, we found this approach to warm starting to be much better than our original strategy of initializing each node randomly.

During early experiments with Branch-&-Bound, it became clear to us that none of the bounding strategies that we have proposed would be able to solve the hardest instances (e.g., $s = 31$ for the data with $n = 63$) in a reasonably short period of time (say, a few hours). In this sense, we cannot claim that optimizing the masked spectral bound is the "silver bullet" for solving extremely large and difficult instances. (It is worth mentioning that we were able to find and prove optimality for the $(n, s) = (63, 31)$ problem in 21 hours using Algorithm 1 (and 56 hours using Algorithm 2) on a Pentium 4 2.4 GHz running Linux. This problem is out of reach using a fixed mask.)

Nevertheless, it was also clear that the various bounding strategies behaved very differently. In order to highlight these differences while maintaining a reasonable testing environment, we settled upon the following testing scheme. For a particular instance of the maximum-entropy sampling problem, we ran each of the four bounding strategies — the three fixed bounding strategies and the optimized bounding strategy — for at most 3600 seconds. We then calculate the gap between the global upper bound (i.e., the largest upper bound of all nodes remaining in the tree) and the global lower bound (i.e., the entropy of the best feasible solution found so far).

Figure 8 gives Branch-&-Bound results for the data with $n = 63$ introduced in §3. The results for each of the four bounding strategies are graphed for $s = 10, \ldots, 50$. From this figure, we can see that the optimization bound strategy did uniformly better than the fixed bound strategies based on the eigenvalue and

Fig. 8. Branch-&-Bound results with $n = 63$ after *3600* seconds for the four bounding strategies. Gap refers to the difference between global upper and lower bounds at termination.



diagonal masks. Furthermore, compared to the one-big spectral partition bound, our method performed better in the ranges $s = 14, \ldots, 19$ and $s = 30, \ldots, 44$.

In Figure 9, we give similar results for a data set with $n = 124$, which was first used in [15]. Again, we see that optimizing the masked spectral bound is quite competitive with the other bounding strategies.

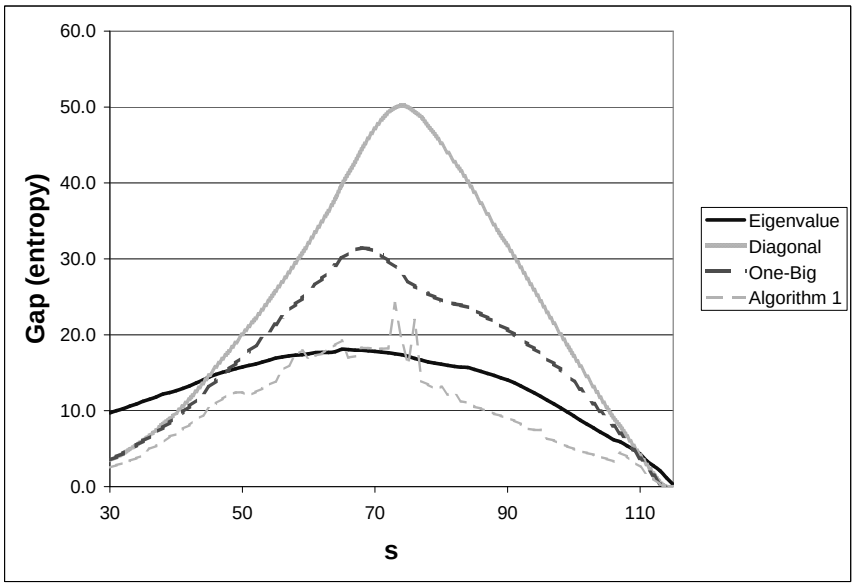
Finally, we mention that we also did some experiments with Algorithm 2 in place of Algorithm 1. However, for reasons that are not entirely clear to us, the performance of Algorithm 2 in Branch-&-Bound was almost uniformly worse than that of Algorithm 1. One possible hypothesis for this behavior is that our warm-start strategy was more favorable for Algorithm 1 than Algorithm 2.

5. Exploiting second-order information

As mentioned at the beginning of §2, one of our initial motivations for considering rank-restricted masks was the opportunity for reducing the dimension of the problem so that second-derivative knowledge of ξ could be incorporated efficiently.

In particular, we were interested in developing a variant of Algorithm 1 with search directions produced by a modified Newton's method with exact Hessians. We did successfully implement just such a method and tested it with small rank, e.g., $k \approx 5$, but unfortunately, it did not outperform the steepest-descent version of Algorithm 1. Downsides to the method included the time required to form the

Fig. 9. Branch-&-Bound results with $n = 124$ after *3600* seconds for the four bounding strategies. Gap refers to the difference between global upper and lower bounds at termination.



Hessian and to factor the modified Hessian as well as the reduced bound quality caused by taking k small. In addition, the strong nonconvexity displayed by ξ resulted in search directions that were not much better than steepest descent.

We also tried other approaches for incorporating second-order information, including BFGS and limited-memory BFGS search directions. Here again, however, our attempts at improving steepest descent were unsuccessful. An added complication was a more stringent strong-Wolfe linesearch required by BFGS-type directions.

We include our experiences here in order to provide a complete picture of our efforts and a starting point for additional research. We also include below the exact Hessian formulae for ξ since these have not appeared before in the literature and since we are hopeful that they will be of use to others. The formulae were developed from [19].

In the results below, all eigenvectors are taken to have unit length. Note that, as in the development of the gradient of $f(V)$, we must define u_l properly when $\lambda_l = \lambda_{l+1}$.

We first examine the Hessian of $\xi_{C,s}(X)$. For simplicity, and with an eye toward implementation, we consider the symmetric variable X to be encoded as a column vector of dimension $n(n+1)/2$, where the lower-triangular part of X is stored in column-major format. In the theorem below, the vectors σ_{kl} are indexed by the entries of X in the same fashion.

Theorem 1. *Suppose $X \succeq 0$, and let $\{(\lambda_k, u_k)\}_{k=1}^n$ be the eigenvalues and eigenvectors of $C \circ X$. Suppose also that $\lambda_s(C \circ X) > \lambda_{s+1}(C \circ X)$. Then ξ is analytic at X with Hessian*

$$\nabla^2 \xi_{C,s}(X) = \sum_{k=1}^{s} \sum_{l=1}^{s} \left(-\frac{1}{\lambda_k \lambda_l} \right) \sigma_{kl} \sigma_{kl}' + 2 \sum_{k=1}^{s} \sum_{l=s+1}^{n} \left(\frac{1}{\lambda_k} \frac{1}{\lambda_k - \lambda_l} \right) \sigma_{kl} \sigma_{kl}',$$

where, for each (k,l),

$$[\sigma_{kl}]_{ij} = \begin{cases} [C \circ u_l u_k']_{ij} + [C \circ u_l u_k']_{ji} & \text{if } i > j \\ [C \circ u_l u_k']_{ij} & \text{if } i = j. \end{cases}$$

We next examine the Hessian of $f(V)$. It is helpful here as well to consider V to be encoded as a column vector in column-major format.

Theorem 2. *Suppose each row of V is nonzero, and define*

$$\begin{aligned} d &:= \operatorname{diag}(VV')^{-1/2}, \\ \bar{V} &:= h(V), \\ X &:= \bar{V}\bar{V}'. \end{aligned}$$

Let $\{(\lambda_k, u_k)\}_{k=1}^n$ be the eigenvalues and eigenvectors of $C \circ X$, and define

$$H := \sum_{k=1}^{s} \lambda_k^{-1} u_k u_k'.$$

Suppose also that $\lambda_s(C \circ X) > \lambda_{s+1}(C \circ X)$. Then f is analytic at V with Hessian

$$\nabla^2 f(V) = \sum_{k=1}^{s} \sum_{l=1}^{s} \left(-\frac{1}{\lambda_k \lambda_l} \right) \sigma_{kl} \sigma_{kl}' + 2 \sum_{k=1}^{s} \sum_{l=s+1}^{n} \left(\frac{1}{\lambda_k} \frac{1}{\lambda_k - \lambda_l} \right) \sigma_{kl} \sigma_{kl}' + \mathcal{M} + \mathcal{N} + \mathcal{P},$$

where, for all (k,l) [here, define $J := C \circ (u_l u_k' + u_k u_l')$ and $\bar{x} := \bar{V}_{\cdot p}$],

$$[\sigma_{kl}]_{ip} = [d \circ (J\bar{x} - \bar{x} \circ (J \circ X)\, e)]_i$$

and where, for all (ip, jq) [here, define $\bar{x} := \bar{V}_{\cdot p}$ and $\bar{y} := \bar{V}_{\cdot q}$],

$$\mathcal{M}_{(ip)(jq)} = 2 \left[dd' \circ \left((e_p' e_q) ee' - (\bar{x} \circ \bar{y})\, e' - e\, (\bar{x} \circ \bar{y})' + \bar{x}\bar{y}' \circ X \right) \circ H \right]_{ij},$$

$$\mathcal{N}_{(ip)(jq)} = \begin{cases} 2 \left[d^2 \circ (3\, \bar{x} \circ \bar{y} \circ (X \circ H)e - \bar{x} \circ H\bar{y} - \bar{y} \circ H\bar{x}) \right]_i & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

and

$$\mathcal{P}_{(ip)(jq)} = \begin{cases} -2 \left[d^2 \circ (X \circ H)e \right]_i & \text{if } i = j \text{ and } p = q \\ 0 & \text{if } i \neq j \text{ or } p \neq q. \end{cases}$$

6. Further directions

One possible extension would be to combine our low-rank approach with gradient-sampling ideas (see [7]) when nondifferentiability becomes an issue at a mask that nearly fathoms, though it may well be that the time would be just as well spent on further children.

We feel that exploiting second-order information, particularly when working with low-rank masks, is still an interesting avenue of research. On some smaller instances (e.g., $n \approx 50$) of Branch-&-Bound with Algorithm 1, we noticed that taking low rank produced overall faster running times than taking full rank. At this time, we are unable to reconcile this behavior with our observations concerning rank in §3, but it would be interesting if one could determine effective strategies for using small rank — and then effectively incorporate second-order information as well.

Acknowledgements. The authors are grateful to Kurt Anstreicher for introducing them to one another and for providing many insightful comments on the paper. Michael Overton provided invaluable input as well.

References

1. Kurt M. Anstreicher, Marcia Fampa, Jon Lee, and Joy Williams, *Continuous relaxations for constrained maximum-entropy sampling*, Integer programming and combinatorial optimization (Vancouver, BC, 1996), Lecture Notes in Computer Science, vol. 1084, Springer, Berlin, 1996, pp. 234–248.
2. ———, *Using continuous nonlinear relaxations to solve constrained maximum-entropy sampling problems*, Mathematical Programming, Series A **85** (1999), no. 2, 221–240.
3. Kurt M. Anstreicher and Jon Lee, *A masked spectral bound for maximum-entropy sampling*, mODa 7—Advances in Model-Oriented Design and Analysis, Contributions to Statistics, Physica-Verlag, Heidelberg, 2004, pp. 1–10.
4. Samuel Burer and Renato D. C. Monteiro, *A projected gradient algorithm for solving the Maxcut SDP relaxation*, Optimization Methods and Software **15** (2001), 175–200.
5. ———, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Mathematical Programming, Series B **95** (2003), no. 2, 329–357.
6. Samuel Burer, Renato D. C. Monteiro, and Yin Zhang, *Rank-two relaxation heuristics for max-cut and other binary quadratic programs*, SIAM Journal on Optimization **12** (2001), no. 2, 503–521.
7. James V. Burke, Adrian S. Lewis, and Michael L. Overton, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM Journal on Optimization (to appear) (2003).
8. William F. Caselton and James V. Zidek, *Optimal monitoring networks*, Statistics and Probability Letters **2** (1984), 129–178.
9. Peter Guttorp, Nhu D. Le, Paul D. Sampson, and James V. Zidek, *Using entropy in the redesign of an environmental monitoring network*, Tech. Report 116, Department of Statistics, University of British Columbia, 1992.
10. Alan Hoffman, Jon Lee, and Joy Williams, *New upper bounds for maximum-entropy sampling*, MODA 6 — Advances in model-oriented design and analysis (Anthony C. Atkinson, Peter Hackl, and Werner G. Müller, eds.), Springer, 2001, pp. 143–153.
11. Chun-Wa Ko, Jon Lee, and Maurice Queyranne, *An exact algorithm for maximum entropy sampling*, Operations Research **43** (1995), no. 4, 684–691.
12. Jon Lee, *Constrained maximum-entropy sampling*, Operations Research **46** (1998), no. 5, 655–664.

13. _____, *Semidefinite programming in experimental design*, Handbook of Semidefinite Programming (Romesh Saigal Henry Wolkowicz and Lieven Vandenberghe, eds.), International Series in Operations Research and Managemant Science, vol. 27, Kluwer, Boston, 2000, pp. 528–532.
14. _____, *Maximum-entropy sampling*, Encyclopedia of Environmetrics (Abdel H. El-Shaarawi and Walter W. Piegorsch, eds.), vol. 3, Wiley, 2001, pp. 1229–1234.
15. Jon Lee and Joy Williams, *A linear integer programming bound for maxim-entropy sampling*, Mathematical Programming, Series B **94** (2003), no. 2-3, 247–256.
16. Werner G. Müller, *Collecting spatial data*, revised ed., Contributions to Statistics, Physica-Verlag, Heidelberg, 2001, Optimum design of experiments for random fields.
17. Paola Sebastiani and Henry P. Wynn, *Maximum entropy sampling and optimal Bayesian experimental design*, Journal of the Royal Statistical Society, Series B (Statistical Methodology) **62** (2000), no. 1, 145–157.
18. Michael C. Shewry and Henry P. Wynn, *Maximum entropy sampling*, Journal of Applied Statistics **46** (1987), 165–170.
19. Nam-Kiu Tsing, Michael K. H. Fan, and Erik I. Verriest, *On analyticity of functions involving eigenvalues*, Linear Algebra and its Application **207** (1994), 159–180.
20. Shiying Wu and James V. Zidek, *An entropy based review of selected NADP/NTN network sites for 1983-86*, Atmospheric Environment **26A** (1992), 2089–2103.