RC22892 (W0309-026) September 5, 2003
Mathematics

IBM Research Report

A Masked Spectral Bound for Maximum-Entropy Sampling

Kurt Anstreicher
University of Iowa
Iowa City, IA

Jon Lee
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

A masked spectral bound for maximum-entropy sampling

Kurt Anstreicher[1] and Jon Lee[2]

[1] Tippie College of Business, University of Iowa, Iowa City, IA. U.S.A.
kurt-anstreicher@uiowa.edu

[2] IBM T.J. Watson Research Center, Yorktown Heights, NY. U.S.A.
jonlee@us.ibm.com

Summary. We introduce a new "masked spectral bound" for the maximum-entropy sampling problem. This bound is a continuous generalization of the very effective "spectral partition bound." Optimization of the masked spectral bound requires the minimization of a nonconvex, nondifferentiable objective over a semidefiniteness constraint. We describe a nonlinear affine scaling algorithm to approximately minimize the bound. Implementation of the procedure obtains excellent bounds at modest computational expense.

Key words: maximum-entropy sampling, experimental design, semidefinite programming, spectral partition bound

Introduction

Let C be a symmetric positive definite matrix of order n. Let s be an integer satisfying $0 < s < n$. For subsets S and T of $N := \{1, 2, \ldots, n\}$, we 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

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

This problem was introduced in [21], developed from an application standpoint in [7, 23] where it was applied to the design of environmental monitoring networks, and has been studied from the algorithmic perspective rather extensively (see [1, 2, 8, 11, 12, 15] and the surveys [13, 14]). In a typical application, C is a sample covariance matrix obtained from time-series observations of one variable at n locations, and it is desired to choose s locations from which to conduct subsequent data collection. The use of entropy as a selection criterion, together with the assumption that values at the n locations are drawn from a multivariate normal distribution, then leads naturally to MESP. (Note that $\ln \det C[S, S]$ is, up to constants, the entropy of the Gaussian random variables having covariance matrix $C[S, S]$.)

Exact algorithms to compute a maximum-entropy design use the "branch-and-bound" framework. Besides lower-bounding heuristics used to find good candidate solutions, a key ingredient is the upper-bounding method. A fast method that can provide a reasonably sharp upper bound on $z(C, n, s)$ is critical to the success of such an approach. Much of the aforementioned algorithmic work concentrates on developing effective upper bounds. The present article continues in this direction.

The organization of the paper is as follows. In Section 1 we describe the masked spectral bound, which is derived using Oppenheim's inequality (see [18]). The new bound is a generalization of the spectral partition bound of [8], which itself is a generalization of both the eigenvalue bound of [11] and the diagonal bound of [8]. Optimization of the masked spectral bound requires the minimization of a nonconvex, nondifferentiable objective over a semidefiniteness constraint. In Section 2 we consider the application of a nonlinear affine scaling algorithm to approximately optimize the masked spectral bound. In Section 3 we give computational results. We find that the performance of the masked spectral bound is superior to several known bounds, and the computational expense to obtain the bound is quite reasonable. Finally, in Section 4 we describe some alternative bounds based on Oppenheim's inequality.

NOTATION: I is an identity matrix; E is a matrix of all ones; e is a vector of all ones; det is determinant; for a matrix X, $\mathrm{diag}(X)$ is the vector of diagonal entries in X; for a vector x, $\mathrm{Diag}(x)$ is the diagonal matrix such that $x = \mathrm{diag}(\mathrm{Diag}(x))$; for a matrix (or vector) X, we denote its transpose by X'; $X \succeq 0$ denotes that X is symmetric and positive semidefinite; $\circ$ is Hadamard (i.e., element-wise) product; $A \bullet B := \mathrm{trace}(AB')$; $\lambda_l(X)$ is the l-th greatest eigenvalue of X. Other basics concerning matrix algebra can be found in [9, 10].

1 The masked spectral bound

A *mask* is an $X \succeq 0$ having $\mathrm{diag}(X) = e$. We define the associated *masked spectral bound* as

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

Special cases include the *diagonal bound* $\xi_C(I)$, the *eigenvalue bound* $\xi_C(E)$, and the *spectral partition bound* $\xi_C(X)$, where X is a block-diagonal matrix with diagonal blocks being matrices of all 1's. Obviously the spectral partition bound is a generalization of the diagonal and eigenvalue bounds, and the masked spectral bound is a further generalization.

Validity of the masked spectral bound ($\xi_C(X) \geq z(C, s, n)$) is based on (i) *Oppenheim's inequality* (see [18])

$$\det A \leq \frac{\det A \circ B}{\prod_{j=1}^{n} B_{jj}} ,$$

where $A \succeq 0$ and $B \succeq 0$ (see [9, Theorem 7.8.6]), and (ii) the eigenvalue inequalities $\lambda_l(A) \geq \lambda_l(B)$, where $A \succeq 0$, and B is a principal submatrix of A (see [9, Thorem 4.3.15]).

We would like to minimize $\xi_C(X)$ over all masks. While the set of masks is indeed a convex set, the function $\xi_C(\cdot)$ is not convex. So, we will have to be content with heuristics that seek a good local minimum of $\xi_C(\cdot)$.

By exploiting 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]).

There are other bounds that do not fit neatly into the present framework. The "linear integer programming bound" introduced in [15] is a strengthening of the spectral partition bound. Although quite effective, that bound is computationally very intensive. The "nonlinear-programming bound" of [1, 2] is based on a concave, continuous relaxation of MESP. That bound, while quite effective on some problems, is somewhat inconsistent and requires parameter tuning that is not completely understood.

2 The minimization method

For $\tilde{X} \succeq 0$, let $u_l(C \circ \tilde{X})$ be an eigenvector, of Euclidean norm 1, associated with $\lambda_l(C \circ \tilde{X})$. Then, as long as $\lambda_s(C \circ \tilde{X}) > \lambda_{s+1}(C \circ \tilde{X})$, we have that the gradient of $\xi_C(\cdot)$ at $\tilde{X}$ is the matrix

$$\nabla_X \xi_C(\tilde{X}) = C \circ \sum_{l=1}^{s} \lambda_l(C \circ \tilde{X}) u_l(C \circ \tilde{X}) u_l(C \circ \tilde{X})' \ .$$

This can be derived using standard results concerning symmetric functions of eigenvalues (see [22], for example). Note that we must define the u_l properly when $\lambda_l(C \circ \tilde{X}) = \lambda_{l+1}(C \circ \tilde{X})$ 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.

When $\lambda_s(C \circ \tilde{X}) = \lambda_{s+1}(C \circ \tilde{X})$, $\xi_C(\cdot)$ is not differentiable at $\tilde{X}$. Thus the problem of finding an optimal mask corresponds to minimizing a nondifferentiable, nonconvex function over a semidefiniteness constraint. There is at present very little literature on problems of this type. A number of recent papers have considered the extension of methods for general nonlinear programming (NLP) to include semidefiniteness (or "Linear Matrix Inequality" (LMI)) constraints. Methodologies based on primal-dual algorithms, augmented Lagrangians, and sequential quadratic programming are described in [3], [4] and [5], respectively. The difficulty with such methods in our context is that they are likely to fail numerically, due to the lack of differentiability of the objective.

An alternative approach to our problem would be to use a general method for the unconstrained minimization of a nondifferentiable function, for example the well-known Bundle-Trust (BT) algorithm (see [20]). To represent the problem of minimizing $\xi_C(\cdot)$ in the form required by such an algorithm, consider the function $\text{svec}(\cdot) : \Re^{n \times n} \to \Re^{n(n-1)/2}$ that takes the superdiagonal components of a matrix X and "stacks" them into a vector x:

$$x = \text{svec}(X) = (X_{12}, X_{13}, \ldots, X_{1n}, X_{23}, X_{24}, \ldots, X_{2n}, \ldots X_{n-1,n})'.$$

Similarly let $X = \mathrm{Smat}(x)$ be the symmetric matrix, with unit diagonal, such that $\mathrm{svec}(\mathrm{Smat}(x)) = x$. The problem of obtaining an optimal mask can then be written in the form

$$\mathrm{MOP}: \qquad \min\ \Big\{\xi_{\mathrm{C}}(\mathrm{Smat(x)}) - \rho \min\{\lambda_{\mathrm{n}}(\mathrm{Smat(x)}), 0\}\Big\},$$

where ρ is a sufficiently large penalty parameter.

The original objective $\xi_C(\cdot)$ could be exponentiated in MOP to extend the domain of definition to include indefinite matrices. However, for ρ sufficiently large, the use of $\xi_C(\cdot)$ should present no difficulties unless s is close to n, in which case the complementary problem could be considered.

From the standpoint of nondifferentiable optimization (NDO), the main difficulty with MOP is the number of variables, $m = n(n-1)/2$. For example, $n = 60$ gives $m = 1770$, and $n = 120$ gives $m = 7140$. In addition, an algorithm for general NDO, such as the BT method, knows nothing of the special structure that underlies MOP and instead treats the objective function to be minimized as a “black box.”

One recently developed method for NDO (see [19]) was designed more specifically for minimizing nondifferentiable functions of the eigenvalues of a matrix, such as $\xi_C(\cdot)$. However, in the control applications for which this algorithm was designed the number of variables is typically small, although the matrices that are functions of the controls can be much larger. For a problem with m variables, the algorithm of [19] samples the gradient at m points in a neighborhood of the current iterate, and then obtains a search direction via minimization of a convex quadratic function of m nonnegative variables. For the dimensions m arising in our application, this work per iteration could be prohibitively large.

Our approach to attempt to minimize $\xi_C(\cdot)$ is based on a heuristic adaptation of the well-known affine scaling algorithm for linear programming (LP). For a given $\tilde{X} \succ 0$ with $\mathrm{diag}(\tilde{X}) = e$, let $G = \nabla_X \xi_C(\tilde{X})$, and consider the linear semidefinite programming problem

$$\begin{aligned} \mathrm{SDP}: \qquad & \min\ G \bullet X \\ & \text{s.t. } \mathrm{diag}(X) = e \\ & \qquad X \succeq 0. \end{aligned}$$

The affine scaling direction for SDP (see [6]) is based on minimizing the linear objective over the “Dikin ellipsoid” induced by the barrier $-\ln \det X$, intersected with the linear equality constraints. For the constraints $\mathrm{diag}(X) = e$, it is straightforward to show (see for example [17]) that the affine scaling direction D at $\tilde{X}$ is given by

$$D := \tilde{X}\big(G - \mathrm{Diag}(u)\big)\tilde{X},$$

where $u = (\tilde{X} \circ \tilde{X})^{-1}\,\mathrm{diag}(\tilde{X}G\tilde{X})$. Given the direction D, we consider a step of the form

$$X := \tilde{X} - \alpha\beta^k D,$$

where $0 < \beta < 1$, and the initial step parameter α corresponds to a fixed fraction of either a “short step” or a “long step.” The short step is based on the limit of the Dikin ellipsoid that is used to define D, while the long step is based on the limit of the true feasible region $X \succeq 0$; see [17, Section 2] for details. We attempt a step with $k = 0$, and we accept the resulting X if $\xi_C(X) < \xi_C(\tilde{X})$. If not, we “backtrack” by incrementing k a limited number of times in an attempt to decrease $\xi_C(\cdot)$. For the highest allowed k, we accept X even if $\xi_C(X) > \xi_C(\tilde{X})$. In our implementation we use $\beta = 1/3$ and $k \leq 2$.

The above strategy for minimizing $\xi_C(\cdot)$ is clearly quite heuristic. For example, due to the nondifferentiability of $\xi_C(\cdot)$, we cannot insure descent on every iteration. However, the use of non-improving steps in algorithms for NDO (for example the well-known subgradient method) is quite common. Even in the case when G is constant, the affine scaling method for SDP (either short-step or long-step) may fail to converge to a solution (see [17]). However, the affine scaling method for LP has been very successful in practice, and it is usually applied with very long steps even though it is known that theoretically the algorithm with such step-sizes could fail to converge (see [16]).

3 Computational results

We have implemented the affine scaling heuristic for approximately minimizing $\xi_C(\cdot)$ in MATLAB. In Figure 1, we illustrate the performance of the algorithm on a problem with $n = 63$, $s = 31$. The data for this example, which has previously been used to evaluate other bounding schemes for $z(C, s, n)$, comes from an environmental monitoring application (see [7]). The algorithm is applied to the complementary problem, and is initialized at a matrix $X_1 = .975E + .025I$ (recall that $X = E$ corresponds to the eigenvalue bound, and $X = I$ corresponds to the diagonal bound). The algorithm is run for 1000 iterations, using an initial step-length α on each iteration corresponding to 50% of a short step. In the figure, we give the gap between the masked spectral bound and the best solution found by a heuristic procedure (the heuristic value is approximately 0.0075 below the optimal value, previously computed using the algorithm of [2]). The gap for the eigenvalue bound ($X_0 = E$)

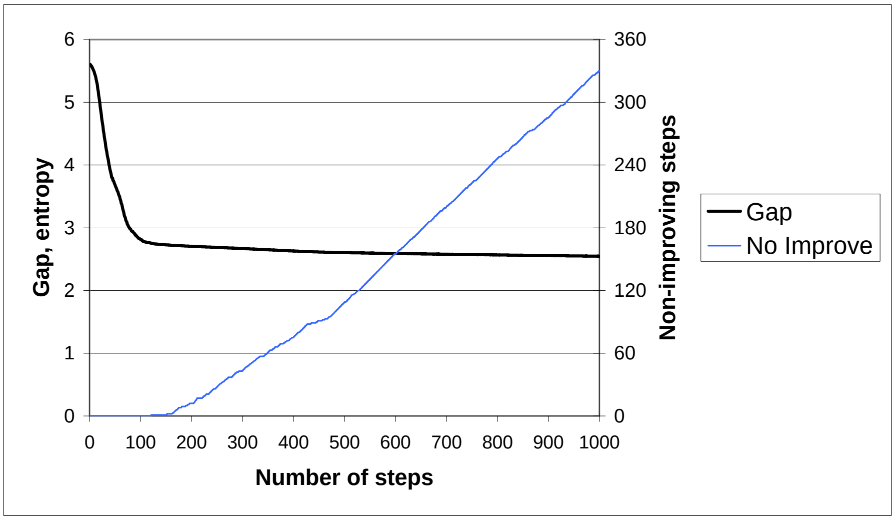

Fig. 1. Decrease in masked spectral bound for problem with $n = 63$, $s = 31$.

is approximately 5.71454. In the figure we also give the cumulative number of non-improving steps. The first 120 iterations are all improving, and decrease the gap to approximately 2.75362, a reduction of almost 52%. The second non-improving step is on iteration 152, after which non-improving steps become more and more frequent. The algorithm nevertheless continues to make gradual progress, eventually decreasing the gap to 2.54639, a reduction of about 55.4% compared to $X_0 = E$. This gap is slightly more than the best gap for the problem obtained in [8]. A substantially smaller gap was computed in [15]. However, the better gaps obtained in [8, 15] are obtained using significantly more computational effort. On a 400 MHz Windows PC the first 120 iterations of the affine scaling method require about 14 seconds, and all 1000 iterations require a total of about 240 seconds. It is to be expected that non-improving iterations are more time-consuming due to backtracking of the steplength. It should also be noted that our implementation is not at all optimized for efficiency, since our primary goal is to evaluate the potential strength of the masked spectral bounds rather than how quickly the bounds can be obtained.

The performance illustrated in Figure 1 appears to be typical for our method. After an initial number of improving steps the algorithm performs non-improving steps more and more frequently, but still obtains gradual decrease in the bound. In our code the iteration sequence is terminated if 50 consecutive steps fail to produce an improvement in the best bound, but using a maximum of 1000 steps this early termination criterion is rarely satisfied.

To evaluate our methodology we have attempted to minimize the masked spectral bound for a number of different test problems. In Figure 2 we give results based on the environmental monitoring data set with $n = 63$ from [7]. For each $s = 3, 4, \ldots, 60$ we give the gaps between several different bounds and the highest

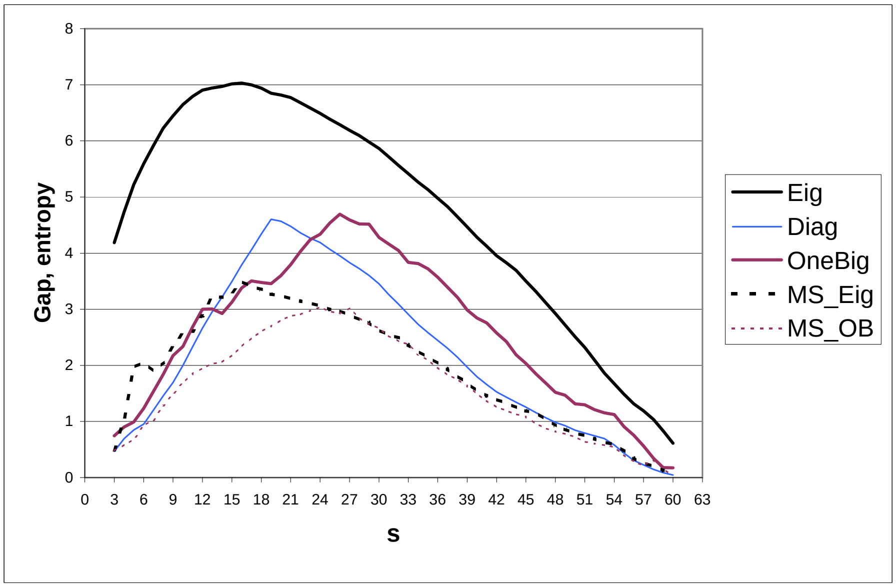

Fig. 2. Comparison of bounds on problems with $n = 63$.

entropy found using a heuristic procedure. In the figure 'Eig' and 'Diag' denote the eigenvalue and diagonal bounds, and 'OneBig' denotes the bound obtained using an X having a single block of ones corresponding to the best solution found by the primal heuristic, and all remaining off-diagonal components zero. 'MS_Eig' and 'MS_OB' refer to the masked spectral bounds computed using an initial X_0 equal to the X that gives the Eig and OneBig bounds, respectively. In both cases the iterative sequence for minimizing $\xi_C(\cdot)$ was initiated at $X_1 = .95X_0 + .05(I + E)$, and the algorithm was run for 1000 iterations. For all of the bounds except Eig we computed bounds based on the original and complementary problems, and report the better of the two. (The original bound was better for s less than 20, 26, 16 and 25 for Diag, OneBig, MS_Eig and MS_OB, respectively.) It is clear from the figure that the masked spectral bound performs very well on this data compared to the previously described Eig, Diag and OneBig bounds. The substantial reduction in the gap comparing MS_Eig to Eig and the excellent performance of MS_OB are both noteworthy.

We also considered problems based on another data set with $n = 124$. In [15] this data was used to evaluate a number of bounds for $z(C, s, n)$, including the Eig and Diag bounds and several variants of the NLP-based bounds from [1, 2]. In Figure 3 we give the gaps between different bounds and a heuristic solution, for $s = 10, 20, \ldots, 120$. For all of the bounds considered in [15], the original problem gave a better bound than the complementary problem for $s \leq 70$. Consequently each bound considered here was based on the original problem for $s \leq 70$, and the complementary problem for $s \geq 80$. The MS_Eig and MS_OB bounds were computed using 500 iterations of the affine scaling procedure, requiring up to 800 seconds per instance on a 400 MHz Windows PC. Among all of the bounds considered in [15],

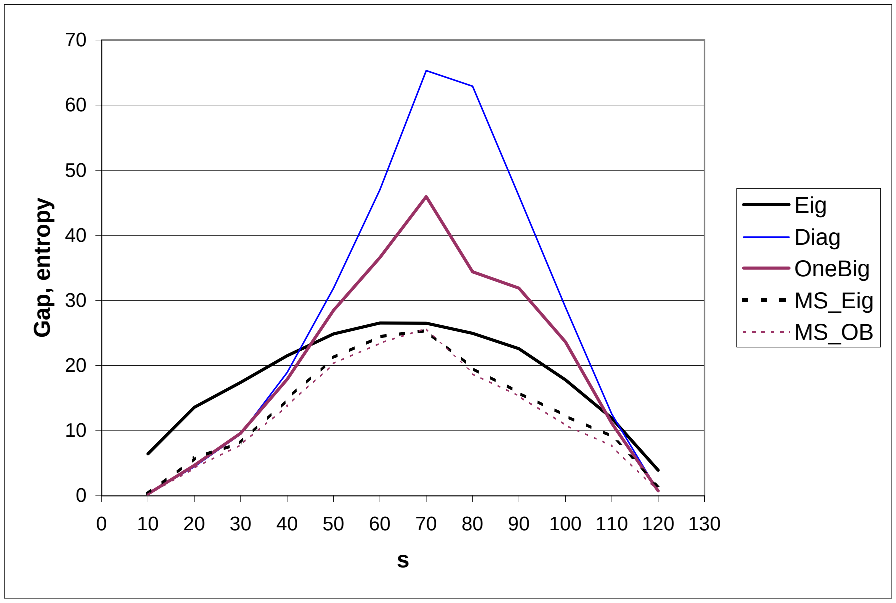

Fig. 3. Comparison of bounds on problems with $n = 124$.

Diag gives the best bound for $s \leq 40$, and Eig gives the best bound for $50 \leq s \leq 110$. For $s = 120$ a partition bound based on blocks of size two is slightly better than Diag. As shown in [15, Table 1.2] the performance of the different NLP bounds is quite poor on these problems. Considering the bound values from [15] together with Figure 3, we conclude that the MS_OB bound is superior to all previously computed bounds on these problems for $10 \leq s \leq 120$. Unfortunately the gaps for intermediate values of s (say $40 \leq s \leq 100$) are still quite high — probably too high to permit the computation of an optimal solution using branch-and-bound in a reasonable amount of time. It is important to note, however, that the optimal values for these problems are unknown, and a substantial fraction of the gap could be due to suboptimality of the lower bound provided by the heuristics.

4 Alternative use of Oppenheim's inequality

Not bothering with the logarithms in this section, the bound that we seek is

$$P_v : \qquad v := \min \left\{ \prod_{l=1}^{s} \lambda_l(C \circ X) \;:\; X \succeq 0,\ \mathrm{diag}(X) = e \right\}.$$

However, by arguing a bit differently we can obtain an alternative bound

$$P_u : \qquad u := \min \left\{ \prod_{l=1}^{s} \lambda_l(C \circ X) / \prod_{l=1}^{s} \mathrm{diag}_{[l]}(X) \;:\; X \succeq 0 \right\},$$

where $\mathrm{diag}_{[l]}(X)$ denotes the l-th least component of $\mathrm{diag}(X)$. Another variation is the bound

$$P_w : \qquad w := \min \left\{ \prod_{l=1}^{s} \lambda_l(C \circ \hat{X}) \;:\; X \succeq 0,\ \hat{X}_{ij} := X_{ij} / \sqrt{X_{ii} X_{jj}} \right\}.$$

All of these are upper bounds on $\max\{\det C[S,S] \;:\; |S| = s\}$. Even though we do not know algorithms to calculate these bounds exactly, it is potentially useful to know the relationship between them.

Theorem 1. $u \leq v = w$.

Proof. Certainly we have $u \leq v$ since every X that is feasible for P_v is feasible for P_u and has the same objective value. Also, we have $w \leq v$ since every X that is feasible for P_v has $\hat{X} = X$ in P_w.

Next, we observe that for every X that is feasible for P_w, $\hat{X}$ is just the Hadamard product of X with $Y := y'y$, where

$$y_i := 1/\sqrt{X_{ii}} \ .$$

It is immediate that $\mathrm{diag}(\hat{X}) = e$ and that $Y \succeq 0$. Furthermore, by the Schur Product Theorem (see [10]), we have that $\hat{X} \succeq 0$. Therefore $\hat{X}$ is feasible for P_v. Finally, the objective value of $\hat{X}$ in P_v is the same as that of X in P_w. Therefore, we have established that $v \leq w$. □

Based on Theorem 1, it might be profitable to attempt to solve P_u instead of P_v to obtain a tighter bound for MESP. However, our limited experience indicates that the objective of P_u (or its logarithm) is substantially more difficult to minimize than that of P_v, due to the presence of the additional nondifferentiable term in the denominator. A minimization framework that deals more carefully with nondifferentiability of the objective might overcome this added difficulty and obtain better bounds. On the other hand we believe that it is likely, but cannot prove, that in the solution of P_u many of the components of $\mathrm{diag}(X)$ are equal.

5 Conclusion

The spectral partition bound of [8] and the related integer linear programming bound of [15] yield significant improvements over spectral bounds based on pre-defined combinatorial masks (for example the eigenvalue bound $\xi_C(E)$ or the diagonal bound $\xi_C(I)$). However, these bounds are potentially too computationally intensive to implement within branch-and-bound due to the effort required to perform the required combinatorial local search. In this paper we have shown, using the new masked spectral bound, that significant improvements over a pre-defined mask can also be achieved through methods of nonlinear programming. Moreover, the required optimization in this case does *not* suffer from the combinatorial explosion associated with local search. We therefore believe that the masked spectral bound is an excellent candidate to enhance the performance of branch-and-bound algorithms for MESP, for example the algorithm based on eigenvalue bounds described in [11].

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 Comput. Sci., vol. 1084, Springer, Berlin, 1996, pp. 234–248.
2. ———, *Using continuous nonlinear relaxations to solve constrained maximum-entropy sampling problems*, Mathematical Programming **85** (1999), no. 2, Ser. A, 221–240.
3. Hande Y. Benson and Robert J. Vanderbei, *Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming*, Mathematical Programming, Series B **95** (2003), 279–302.
4. Bassem Fares, Pierre Apkarian, and Dominikus Noll, *An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory*, International J. Control **74** (2001), 348–360.
5. Bassem Fares, Dominikus Noll, and Pierre Apkarian, *Robust control via sequential semidefinite programming*, SIAM J. Control Optim. **40** (2002), 1791–1820.
6. Leonid Faybusovich, *Dikin's algorithm for matrix linear programming problems*, Lecture notes in control and information sciences **197** (1994), 237–247.
7. 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, The Department of Statistics, The University of British Columbia, 1992.

8. Alan Hoffman, Jon Lee, and Joy Williams, *New upper bounds for maximum-entropy sampling*, mODa 6—Advances in model-oriented design and analysis (Puchberg/Schneeberg, 2001), Contrib. Statist., Physica, Heidelberg, 2001, pp. 143–153.
9. Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
10. ______, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.
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 (Henry Wolkowicz, Romesh Saigal and Lieven Vandenberghe, eds.), International Series in Operations Research and Management Science, vol. 27, Kluwer Acad. Publ., Boston, MA, 2000, pp. 528–532.
14. ______, *Maximum-entropy sampling*, Encyclopedia of Environmetrics (Abdel H. El-Shaarawi and Walter W. Piegorsch, eds.), vol. 3, John Wiley & Sons Inc., 2001, pp. 1229–1234.
15. Jon Lee and Joy Williams, *A linear integer programming bound for maximentropy sampling*, Mathematical Programming **94** (2003), no. 2-3, Ser. B, 247–256.
16. Walter F. Mascarenhas, *The affine scaling algorithm fails for $\lambda = .999$*, SIAM J. Optim. **7** (1997), 34–46.
17. Masakazu Muramatsu, *Affine scaling algorithm fails for semidefinite programming*, Mathematical Programming **83** (1998), 393–406.
18. Alexander Oppenheim, *Inequalities connected with definite Hermitian forms*, Journal of the London Mathematical Society **5** (1930), 114–119.
19. Michael Overton, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, Presentation, 18th International Symposium on Mathematical Programming, Copenhagen, August 2003.
20. Helga Schramm and Jochem Zowe, *A combination of the bundle approach and trust region concept*, Advances in Mathematical Optimization (J. Guddat et al., ed.), Akademie Verlag, Berlin, 1988.
21. Michael C. Shewry and Henry P. Wynn, *Maximum entropy sampling*, Journal of Applied Statistics **46** (1987), 165–170.
22. Nam-Kiu Tsing, Michael K. H. Fan, and Erik I. Verriest, *On analyticity of functions involving eigenvalues*, Linear Algebra and its Applications **207** (1994), 159–180.
23. 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.