On Algorithmic Advances for Maximum-Entropy Sampling

by

Zhongzhu Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2024

Doctoral Committee:

Professor Marcia Fampa, Co-Chair
Professor Jon Lee, Co-Chair
Professor Nikhil Bansal
Professor Albert S. Berahas

Zhongzhu Chen
zhongzhc@umich.edu
ORCID iD: 0000-0003-4998-4293

ACKNOWLEDGEMENTS

During my Ph.D. journey, I have been incredibly fortunate to receive support and help from wonderful advisors, collaborators, colleagues, family, and friends. Initially, I want to express my profound gratitude to Prof. Jon Lee, my advisor, and Prof. Marcia Fampa, my co-advisor. They granted me the opportunity to embark on my journey in optimization, provided continuous guidance in my academic research, and offered invaluable assistance in life decisions and job seeking. I wouldn't have grown as a researcher without their inspiring advice and steadfast support. The passion for research and the charisma of Prof. Lee and Prof. Marcia Fampa have motivated me to persist in my research in optimization, and their influence will profoundly shape my future.

In addition to my advisor, I would like to extend my gratitude to my committee members, Prof. Nikhil Bansal and Prof. Albert Berahas, for their invaluable participation and suggestions throughout my Ph.D. journey. My sincere thanks also go to all the professors at the University of Michigan – Prof. Xiuli Chao, Prof. Marina Epelman, Prof. Viswanath Nagarajan, Prof. Siqian Shen, and others – from whom I have had the privilege of learning. Their insights have significantly broadened my knowledge and strengthened my understanding of research. I am equally grateful to all the staff at the University of Michigan for their immense support and for making my research life much more convenient. Additionally, I would like to offer a special thank you to Prof. Xiuli Chao for introducing me to the University of Michigan in the summer of 2018, a pivotal moment in my academic journey.

During my time at the University of Michigan, I have had the privilege of working with many wonderful researchers: Prof. Anima Anandkumar, Prof. Amélie Lambert, Prof. Bo Li, Prof. Chaowei Xiao, Prof. Dawn Song, Prof. Huan Zhang, Prof. Mingyan Liu, Prof. Xueru Zhang, Prof. Yang Liu, Dr. Kun Jin, Dr. Weili Nie, Jiongxiao Wang, Jiawei Zhang, Tongxin Yin. I am deeply grateful for your insights, from which I have learned immensely.

Finally, I must express my profound gratitude to my parents and family for their unconditional love and support. Thank you for always having my back, believing in me unwaveringly regardless of the decisions I made or the paths I chose.

In addition, the work was supported in part by ONR grant N00014-17-1-2296, AFOSR grants FA9550-19-1-0175 and FA9550-22-1-0172, and a University of Michigan Rackham Predoctoral Fellowship.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

The maximum-entropy sampling problem (MESP) is a fundamental and challenging combinatorial-optimization problem, at the intersection of information theory, machine learning, and optimization. The goal is to find a maximum-entropy subset of s random variables, from a universe of n correlated Gaussian random variables, which is a means of choosing the s-subset with maximum information. MESP finds application in experimental design (Shewry and Wynn, 1987), spatial statistics (Zidek, Sun, and Le, 2000), financial portfolio selection (Bera and Park, 2008), feature selection (Song and Liò, 2010), active learning (Qiu, Miller, and Kesidis, 2016), and many references in (Fampa and Lee, 2022, Chapter 4).

MESP is an NP-hard problem. Research in this area often focuses on exact algorithms within a branch-and-bound (B&B) framework, as introduced by (Ko, Lee, and Queyranne, 1995). This framework involves the implicit enumeration of potential solutions while maintaining upper and lower bounds on the optimal value of MESP to efficiently discard non-optimal solutions. Consequently, the solution speed of the B&B method heavily relies on the tightness of these bounds. In this dissertation, we propose enhanced upper- and lower-bounding techniques to expedite the branch-and-bound framework when applied to MESP, facilitating the handling of large-size instances. Our approach includes a "mixing" methodology for combining multiple convex-relaxation upper bounds of MESP to derive superior bounds. We also present a generalization of upper bounds from (Nikolov, 2015; Li and Xie, 2023), which turns out to be the best for many instances. Moreover, we introduce a "general scaling" technique for reducing the integrality gap further, compared to one of the most powerful techniques, "scaling" (Anstreicher, Fampa, Lee, and Williams, 1996). We also address the theoretical void in the "masking" technique, a promising method without much exploration. Additionally, we introduce an efficient, limited-memory quasi-Newton algorithm for finding nearly optimal masks.

CHAPTER 1

Introduction

1.1 Literature review for MESP and CMESP

The *maximum-entropy sampling problem*, a fundamental problem in optimal statistical design, was formally introduced in the "design of experiments" literature by (Shewry and Wynn, 1987) and then applied in many areas such as the re-design of environmental-monitoring networks (Zidek, Sun, and Le, 2000). It aims to find a subset of cardinality s with *maximum information* from a ground set of n continuous random variables. The concept of information can be measured by the so-called *differential entropy*, as introduced in (Shannon, 1948). In the Gaussian case, the problem can be cast as

$$z(C,s) := \max\left\{ \operatorname{ldet} C[S(x),S(x)] \ : \ \mathbf{e}^{\mathsf{T}}x = s,\ x \in \{0,1\}^n \right\}. \tag{MESP}$$

where ldet denotes the natural logarithm of the determinant, C is an $n \times n$ covariance matrix (of Gaussian random variables), $s < n$ is a positive integer with $0 < s \leq \operatorname{rank}(C)$ so that MESP always has a feasible solution with finite objective value, and $S(x)$ is a subset of $\{1,2,\ldots,n\}$ with support vector x. $C[S(x),S(x)]$ denotes the submatrix of C having rows and columns indexed by $S(x)$. We further assume that $C[j,j] > 0$ for all $j \in N$, because if we had any $C[j,j] = 0$, then such a j could not be in any feasible solution of MESP having objective value greater than $-\infty$. In the constrained version CMESP, we also have $m \geq 0$ side constraints: $Ax \leq b$ where $A \in \mathbb{R}^{m\times n}$

$$z(C,s,A,b) := \max\left\{ \operatorname{ldet} C[S(x),S(x)] \ : \ \mathbf{e}^{\mathsf{T}}x = s,\ x \in \{0,1\}^n,\ Ax \leq b \right\}. \tag{CMESP}$$

In this work, we will use CMESP when introducing the upper bounds and the techniques for completeness. While elsewhere, we will choose to work with MESP when the linear constraints do not matter, so as to for simplify notation.

In the environmental-monitoring application of MESP, we collect time-series observations

from n environmental monitoring stations, and we prepare a sample covariance matrix C, see (Al-Thani and Lee, 2020a,b), for details on how this can be done effectively. In many situations, keeping all n of the monitoring stations running is too costly, and so we wish to select a subset of size s, and continue monitoring only at them. Maximizing the "differential entropy" of the s sites is a means of choosing the s-subset with maximum information. Take, for instance, the National Acidic Deposition Program's (NADP) National Trends Network (NTN) (NADP, 2018), which monitors precipitation chemistry across the United States. Out of 379 NTN monitoring sites, 255 are currently active, chosen based on spatial distribution criteria to ensure even coverage. However, by integrating this selection process into the MESP framework, it can be discovered that the current spatial distribution approach is not the most effective. The optimal solution to MESP suggests a better allocation of active sites, such as removing a less impactful site in western Tennessee in favor of multiple sites in northern Colorado. This optimized selection does not increase costs but significantly enhances the breadth and immediacy of pollution data collection.

In finance, we want to maximize the degree of portfolio diversification, measured by entropy (Hoskisson, Hitt, Johnson, and Moesel, 1993; Bera and Park, 2008; Jana, Roy, and Mazumder, 2007), in a combinatorial Markowitz-style setting, where we want to choose s go/no-go investments from n, subject to a lower limit on yield mean and an upper limit on yield variance. The whole problem can be formulated into MESP. In recommendation systems or search engines, it is crucial to ensure the first few pages feature diverse, information-rich items. This can be achieved by maximizing entropy, which enhances information content while minimizing the recommendation of similar items (Jin, Mobasher, and Zhou, 2005; Qin and Zhu, 2013).

In the context of machine learning and data science, the application of MESP is also extensive. In feature selection processes, machine learning models are typically confronted with a large set of features that are not mutually independent. An increase in the count of features selected for model fitting can enhance the model's accuracy on training data. However, more features may increase the risk of feature collinearity, which escalates the uncertainty in parameter estimation, thus precipitating overfitting and diminishing accuracy on test datasets. A strategy to mitigate these challenges is to formulate feature selection into MESP. This approach can maximize information retention in the chosen features while concurrently minimizing feature collinearity, thus increasing prediction accuracy while curtailing uncertainty and overfitting risks (Basu, Micchelli, and Olsen, 2000; Song and Liò, 2010). Another key application of MESP is its use in active learning, a critical area in the era of big data. For the training of a reliable machine learning model, a large volume of high-quality labeled data is essential. While data is plentiful, high-quality labels are limited to a

smaller subset of this data, and their generation requires costly human effort. It is recognized that not all data equally contribute to the training of a machine learning model. Leveraging a small set of labeled data and the predictions from multiple models trained on this data, we can effectively select the most informative and diverse unlabeled data for human labeling through the formulation of an MESP (Qiu, Miller, and Kesidis, 2016). This method aims to enhance model accuracy with the least additional effort. Additionally, MESP also plays an important role in compressive sensing (Hoch, Maciejewski, Mobli, Schuyler, and Stern, 2014), image sampling (Zilly, Buhmann, and Mahapatra, 2017), and the many references in (Fampa and Lee, 2022, Chapter 4).

Furthermore, MESP serves as a nice example of a "non-factorable" mixed-integer nonlinear program. When C is a diagonal matrix, CMESP reduces to a general cardinality-constrained binary linear program. (Al-Thani and Lee, 2021, 2023) established that when C is tridiagonal (or even when the support graph of C is a spider with a bounded number of legs), MESP is then polynomially solvable by dynamic programming.

Despite the widespread application of MESP in various fields, solving it is generally NP-hard (proved by reduction from the stable set decision problem in (Ko, Lee, and Queyranne, 1995)). MESP was first approached for global optimization by (Ko, Lee, and Queyranne, 1995) and CMESP was first algorithmically approached by (Lee, 1998) and then by (Anstreicher, Fampa, Lee, and Williams, 1996, 1999). The "branch-and-bound" method (B&B) is a key technique for tackling the MESP, as it implicitly enumerates potential solutions while maintaining upper and lower bounds of the optimal value to efficiently discard non-optimal solutions (Fampa and Lee, 2022, chapter 2).

Specifically, the B&B algorithm maintains a list of subproblems of CMESP having the form

$$\begin{aligned} z(C,s,A,b;F_0,F_1) := \max \Big\{ & \operatorname{ldet} C[S(x),S(x)] \ : \ \mathbf{e}^{\mathsf{T}}x = s, \\ & x \in \{0,1\}^n, \ x_i = 0, i \in F_0, \ x_i = 1, i \in F_1, \ Ax \leq b \Big\}, \end{aligned} \tag{CMESP-sub}$$

where F_0 (F_1) is the set of indices fixed into (out of) a potential solution of CMESP. In the context of the B&B algorithm, two primary structures are identified: branching and bounding. The branching mechanism involves selecting a branching index $j \in \{0,1\}^n \backslash F_0 \backslash F_1$, leading to the creation of two child subproblems: $z(C,s,A,b;F_0,F_1+j)$ and $z(C,s,A,b;F_0+j,F_1)$. This process is iterated until subproblems become trivial for enumeration or can be dismissed. Concurrently, the bounding structure plays a crucial role. The global upper bound is defined as the maximum of the upper bounds of all subproblems, while the global lower bound is the maximum of the objective values of all feasible solutions so far (i.e., the

so-called incumbent). Employing bounding techniques, a subproblem is discarded if it is infeasible or if its optimal value's upper bound is lower than the global lower bound. The termination of the B&B algorithm occurs either when the list of subproblems is exhausted or when the gap between the global upper and lower bounds is sufficiently small. In such cases, the feasible solution corresponding to the global lower bound is considered optimal or approximately optimal. Therefore, with effective bounds, we can significantly reduce the total running time of B&B from years to minutes for many practical instances.

For the lower bound, it can be observed that MESP is intrinsically a non-monotone submodular maximization problem with cardinality constraints. The work by (Lee, Mirrokni, Nagarajan, and Sviridenko, 2009/10) presents an approximation algorithm with a $\frac{1}{6}$-ratio, utilizing a local-search approach under the assumption of a consistently positive objective function. Subsequently, (Li and Xie, 2023) enhanced this methodology by introducing a $\min\{s \log s, s \log(n - s - n/s + 2)\}$-gap approximation algorithm, by similarly employing a local-search framework.

For the upper bound, (Ko, Lee, and Queyranne, 1995) introduced the "spectral bound". (Lee, 1998) extended the spectral approach to CMESP. (Anstreicher, Fampa, Lee, and Williams, 1996) and (Anstreicher, Fampa, Lee, and Williams, 1999) developed a bound, the so-called "NLP bound", employing a novel convex relaxation. (Anstreicher, 2018) developed the "BQP bound", using an extended formulation based on the Boolean quadric polytope. (Anstreicher, 2020) introduced the "linx bound", based on a clever convex relaxation. (Nikolov, 2015) gave a novel "factorization bound" based on a subtle convex relaxation. This was further developed by (Li and Xie, 2023). There are also a lot of improvement techniques for upper bounds proposed in the literature, such as "scaling" and "complementation" ((Anstreicher, Fampa, Lee, and Williams, 1996, 1999)) as well as "masking" ((Anstreicher and Lee, 2004)). All of these convex-optimization based bounds admit variable fixing methodology based on convex duality (see (Fampa and Lee, 2022), for example). In computational practice, the best bounds appear to be the linx bound and the NLP bound. The BQP bound is generally too time-consuming to compute. However, the BQP bound can be better than the others in some cases.

My research mainly focuses on enhancing the *upper bounds* for MESP, making many instances solvable in a reasonable time for practical applications.

1.2 The linx, BQP, and NLP upper bounds for CMESP

We present in detail several key upper bounds from the literature here, which are fundamental to our analysis throughout this dissertation.

Linx Bound: The linx bound was first analyzed and developed in (Anstreicher, 2020) (see (Fampa and Lee, 2022, Section 3.3) for more details).

For $x \in [0,1]^n$, we define

$$f_{\text{linx}}(C;x) := \tfrac{1}{2}\left(\text{ldet}\left(C\,\text{Diag}(x)C + \text{Diag}(\mathbf{e}-x)\right)\right)$$

with domain

$$\text{dom}\,(f_{\text{linx}}) := \left\{x \in \mathbb{R}^n : C\,\text{Diag}(x)C + \text{Diag}(\mathbf{e}-x) \succ 0\right\}.$$

We then define the *linx bound*

$$z_{\text{linx}}(C,s,A,b) := \max\left\{f_{\text{linx}}(C,x) \;:\; x \in P_{\text{linx}}(n,s,A,b)\right\}, \tag{linx}$$

where $P_{\text{linx}}(n,s,A,b) := \{\mathbf{e}^\mathsf{T}x = s,\ 0 \le x \le \mathbf{e},\ Ax \le b\}$. We say that x is feasible to linx if x satisfies all the constraints in linx.

BQP Bound: The Boolean-Quadratic-Polytope (BQP) bound was first analyzed and developed in (Anstreicher, 2018) (see (Fampa and Lee, 2022, Section 3.6) for more details). We lift to matrix space, by defining the convex set

$$P_{\text{BQP}}(n,s,A,b) := \left\{(x,X) \in \mathbb{R}^n \times \mathbb{S}^n \;:\; X - xx^\mathsf{T} \succeq 0,\ \text{diag}(X) = x,\ \mathbf{e}^\mathsf{T}x = s,\ X\mathbf{e} = sx,\ Ax \le b\right\}.$$

We define

$$f_{\text{BQP}}(C;x,X) := \text{ldet}\left(C \circ X + \text{Diag}(\mathbf{e}-x)\right),$$

with domain

$$\text{dom}\,(f_{\text{BQP}}) := \left\{(x,X) \in \mathbb{R}^n \times \mathbb{S}^n \;:\; C \circ X + \text{Diag}(\mathbf{e}-x) \succ 0\right\}.$$

The *BQP bound* is defined as

$$z_{\text{BQP}}(C,s,A,b) := \max\left\{f_{\text{BQP}}(C;x,X) \;:\; (x,X) \in P_{\text{BQP}}(n,s,A,b)\right\}. \tag{BQP}$$

We say that x is feasible to BQP if x satisfies all the constraints in BQP.

NLP Bound: The first upper bound based on a convex-relaxation is the "NLP bound" proposed in (Anstreicher, Fampa, Lee, and Williams, 1996, 1999). Let

$$P_{\text{NLP}}(n, s, A, b) := \left\{ x \in \mathbb{R}^n : 0 \le x \le \mathbf{e},\ \mathbf{e}^{\mathsf{T}} x = s,\ Ax \le b \right\}.$$

We now define

$$f_{\text{NLP}}(C, d, p; x) := \operatorname{ldet}\left(\operatorname{Diag}(x^{p/2}) C \operatorname{Diag}(x^{p/2}) + \operatorname{Diag}(d^x - dx^p) \right),$$

where $x^{p/2} := (x_1^{p_1/2}, \ldots, x_n^{p_n/2})$, $d^x - dx^p := (d_1^{x_1} - d_1 x_1^{p_1}, \ldots, d_n^{x_n} - d_n x_n^{p_n})$, and the parameters $d_i > 0$ and $p_i \ge 1$ with domain

$$\operatorname{dom}(f_{\text{NLP}}) := \left\{ x \in \mathbb{R}^n \ : \ \operatorname{Diag}(x^{p/2}) C \operatorname{Diag}(x^{p/2}) + \operatorname{Diag}(d^x - dx^p) \succ 0 \right\}.$$

The *NLP bound* is then defined as

$$z_{\text{NLP}}(C, d, p, s, A, b) := \max \left\{ f_{\text{NLP}}(C, d, p; x) \ : \ (x, X) \in P_{\text{NLP}}(n, s, A, b) \right\}. \tag{NLP}$$

Notice that when $x \in \{0, 1\}^n$, this reduces to $\operatorname{ldet}(\operatorname{Diag}(x) C \operatorname{Diag}(x) + \operatorname{Diag}(\mathbf{e} - x))$; that is, the parameters disappear, giving us again an exact relaxation. But it turns out that the parameters can be chosen to gain concavity on the feasible region (see (Anstreicher, Fampa, Lee, and Williams, 1999) for the very technical details).

Throughout this dissertation, for notations associated with the above upper bounds, such as $f_{\text{linx}}, z_{\text{linx}}, P_{\text{linx}}$, symbols such as C, s, p, d will be omitted for brevity when they are apparent from the context. We may also omit $_{\text{linx}}$, $_{\text{BQP}}$, and $_{\text{NLP}}$ for brevity or saving space for other subscripts when it should not confuse the readers. Moreover, we will also omit A and b when the linear constraints do not matter.

1.3 Key improving techniques for the upper bounds

Besides the upper bounds themselves, we are also interested in various techniques for improving these upper bounds. In the following, we review several such techniques proposed in the literature.

1.3.1 Scaling

The first important general technique for potentially improving some of the entropy upper bounds is "scaling", based on the simple observation that for a positive constant γ, and S with $|S| = s$, we have that $\det(\gamma C)[S,S] = \gamma^s \det C[S,S]$. With this identity, we can easily see that

$$z(C, s, A, b) = z(\gamma C, s, A, b) - s \log \gamma. \tag{scaling}$$

So upper bounds for $z(\gamma C, s, A, b)$ yield upper bounds for $z(C, s, A, b)$, shifted by $-s \log \gamma$. It is important to note that many bounding methods are *not* invariant under scaling; that is, the bound does *not* generally shift by $-s \log \gamma$ (notable exceptions being the spectral and factorization bounds for MESP). Scaling was first introduced in (Anstreicher, Fampa, Lee, and Williams, 1996, 1999), and then exploited in (Anstreicher, 2018, 2020; Al-Thani and Lee, 2020a; Chen, Fampa, Lambert, and Lee, 2021; Chen, Fampa, and Lee, 2022, 2023). Scaling can be seen as a technique aimed at adjusting the shape of concave continuous relaxations of the objective of MESP in order to decrease the gap between the upper bounds and $z(C, s, A, b)$; see (Chen, Fampa, and Lee, 2022) for an exploration of this in the context of the linx bound. In this dissertation, we employ the scaled version of several upper bounds, which are delineated herein.

Scaled linx Bound: We define

$$f_{\text{linx}}(C, s; \gamma; x) := \tfrac{1}{2} \left(\operatorname{ldet} \left(\gamma C \operatorname{Diag}(x) C + \operatorname{Diag}(\mathbf{e} - x) \right) - s \log \gamma \right),$$

with

$$\operatorname{dom} \left(f_{\text{linx}}; \gamma \right) := \left\{ x \in \mathbb{R}^n : \gamma C \operatorname{Diag}(x) C + \operatorname{Diag}(\mathbf{e} - x) \succ 0 \right\}.$$

We then define the *scaled linx bound*

$$z_{\text{linx}}(C, s, A, b; \gamma) := \max \left\{ f_{\text{linx}}(C, s; \gamma; x) \ : \ x \in P_{\text{linx}}(n, s, A, b) \right\}. \tag{scaled linx}$$

Scaled BQP Bound: We define

$$f_{\text{BQP}}(C, s; \gamma; x, X) := \operatorname{ldet} \left(\gamma C \circ X + \operatorname{Diag}(\mathbf{e} - x) \right) - s \log \gamma,$$

with domain

$$\operatorname{dom} \left(f_{\text{BQP}}; \gamma \right) := \left\{ (x, X) \in \mathbb{R}^n \times \mathbb{S}^n \ : \ \gamma C \circ X + \operatorname{Diag}(\mathbf{e} - x) \succ 0 \right\}.$$

The *scaled BQP bound* is defined as

$$z_{\text{BQP}}(C, s, A, b; \gamma) := \max \left\{ f_{\text{BQP}}(C, s; \gamma; x, X) \ : \ (x, X) \in P_{\text{BQP}}(n, s, A, b) \right\}. \tag{scaled BQP}$$

Scaled NLP Bound: We define

$$f_{\text{NLP}}(C, s, d, p; \gamma; x) := \operatorname{ldet}\left(\gamma \operatorname{Diag}(x^{p/2}) C \operatorname{Diag}(x^{p/2}) + \operatorname{Diag}(d^x - dx^p)\right) - s \log \gamma,$$

where $x^{p/2} := (x_1^{p_1/2}, \ldots, x_n^{p_n/2})$, $d^x - dx^p := (d_1^{x_1} - d_1 x_1^{p_1}, \ldots, d_n^{x_n} - d_n x_n^{p_n})$, and the parameters $d_i > 0$ and $p_i \geq 1$ with domain

$$\operatorname{dom}(f_{\text{NLP}}; \gamma) := \left\{ x \in \mathbb{R}^n \ : \ \gamma \operatorname{Diag}(x^{p/2}) C \operatorname{Diag}(x^{p/2}) + \operatorname{Diag}(d^x - dx^p) \succ 0 \right\}.$$

The *scaled NLP bound* is then defined as

$$z_{\text{NLP}}(C, s, d, p, A, b; \gamma) := \max \left\{ f_{\text{NLP}}(C, s, d, p; \gamma; x) \ : \ (x, X) \in P_{\text{NLP}}(n, s, A, b) \right\}. \tag{scaled NLP}$$

1.3.2 Complementation

The second key technique for obtaining bounds is "complementation", first utilized by (Anstreicher, Fampa, Lee, and Williams, 1996, 1999). If C is invertible, we have

$$z(C, s) = z(C^{-1}, n - s) + \operatorname{ldet} C. \tag{MESP-comp}$$

where $z(C^{-1}, n - s)$ denotes the optimal value of MESP with C, s replaced by $C^{-1}, n - s$. Similarly, we also have

$$z(C, s, A, b) = z(C^{-1}, n - s, -A, b - A\mathbf{e}) + \operatorname{ldet} C. \tag{CMESP-comp}$$

where $z(C^{-1}, n-s, -A, b-A\mathbf{e})$ denotes the optimal value of CMESP with C, s, A, b replaced by $C^{-1}, n-s, -A, b-A\mathbf{e}$, respectively. So we have a *complementary* MESP (CMESP) problem and *complementary* bounds (i.e., bounds for the complementary problem plus $\operatorname{ldet} C$) immediately give us bounds on z. Some upper bounds on z also shift by $\operatorname{ldet} C$ under complementing (notably, (Ko, Lee, and Queyranne, 1995; Anstreicher, 2020)), in which case there is no additional value in computing the complementary bound. But other upper bounds ((Anstreicher, Fampa, Lee, and Williams, 1999; Hoffman, Lee, and Williams, 2001; Lee and Williams, 2003; Anstreicher and Lee, 2004; Anstreicher, 2018; Chen, Fampa, and Lee, 2023))

are generally not invariant under complementation. Details on all of this can be found in (Fampa and Lee, 2022).

1.3.3 Masking

The third important technique for potentially improving some of the entropy upper bounds is "masking". Given a positive integer n, a *mask* (also known as a "correlation matrix") is an $n \times n$ symmetric positive-semidefinite matrix with $\operatorname{diag}(M) = \mathbf{e}$. We denote the set of order-n masks as $\mathcal{M}_n$. Masking for MESP, introduced in full generality in (Anstreicher and Lee, 2004), is based on the observation that for any $S \subseteq N$ and mask M, we have $\det C[S,S] \leq \det(C \circ M)[S,S]$. That is, masking cannot decrease entropy. Therefore, for any mask $M \in \mathcal{M}_n$, we have

$$z(C, s, A, b) \leq z(C \circ M, s, A, b). \tag{masking}$$

This implies that upper bounds for $z(C \circ M, s)$ are also upper bounds for $z(C, s)$. In this dissertation, we employ the masked version of several upper bounds, which are delineated herein.

Masked Linx Bound: We define

$$f_{\text{linx}}(C, s; M; x) := \tfrac{1}{2} \left(\operatorname{ldet} \left((C \circ M) \operatorname{Diag}(x) (C \circ M) + \operatorname{Diag}(\mathbf{e} - x) \right) \right),$$

with

$$\operatorname{dom}(f_{\text{linx}}; M) := \left\{ x \in \mathbb{R}^n : (C \circ M) \operatorname{Diag}(x) (C \circ M) + \operatorname{Diag}(\mathbf{e} - x) \succ 0 \right\}.$$

We then define the *masked linx bound*

$$z_{\text{linx}}(C, s, A, b; M) := \max \left\{ f_{\text{linx}}(C, s; M; x) \ : \ x \in P_{\text{linx}}(n, s, A, b) \right\}. \tag{masked linx}$$

Maksed BQP Bound: We define

$$f_{\text{BQP}}(C, s; M; x, X) := \operatorname{ldet} \left((C \circ M) \circ X + \operatorname{Diag}(\mathbf{e} - x) \right),$$

with domain

$$\operatorname{dom}(f_{\text{BQP}}; M) := \left\{ (x, X) \in \mathbb{R}^n \times \mathbb{S}^n \ : \ (C \circ M) \circ X + \operatorname{Diag}(\mathbf{e} - x) \succ 0 \right\}.$$

The *masked BQP bound* is defined as

$$z_{\text{BQP}}(C, s, A, b; M) := \max \left\{ f_{\text{BQP}}(C, s; M; x, X) \ : \ (x, X) \in P_{\text{BQP}}(n, s, A, b) \right\}. \tag{masked BQP}$$

Masked NLP Bound: We define

$$f_{\text{NLP}}(C, s, p, d; M; x) := \operatorname{ldet}\left(\operatorname{Diag}(x^{p/2})(C \circ M)\operatorname{Diag}(x^{p/2}) + \operatorname{Diag}(d^x - dx^p)\right),$$

where $x^{p/2} := (x_1^{p_1/2}, \ldots, x_n^{p_n/2})$, $d^x - dx^p := (d_1^{x_1} - d_1 x_1^{p_1}, \ldots, d_n^{x_n} - d_n x_n^{p_n})$, and the parameters $d_i > 0$ and $p_i \geq 1$ with domain

$$\operatorname{dom}(f_{\text{NLP}}; M) := \left\{ x \in \mathbb{R}^n \ : \ \operatorname{Diag}(x^{p/2})(C \circ M)\operatorname{Diag}(x^{p/2}) + \operatorname{Diag}(d^x - dx^p) \succ 0 \right\}.$$

The *masked NLP bound* is then defined as

$$z_{\text{NLP}}(C, s, p, d, A, b; M) := \max \left\{ f_{\text{NLP}}(C, s, p, d; M; x) \ : \ (x, X) \in P_{\text{NLP}}(n, s, A, b) \right\}. \tag{masked NLP}$$

1.4 Notations

We introduce the notations that will be used throughout this work in Tables 1.1 and 1.2. In the following, we will omit n when it is clear from the context.

Notation	Description
det	determinant
log	natural logarithm
ldet	natural logarithm determinant
$\subseteq$	subset
$\circ$	Hadamard (i.e., element-wise) product
$A \bullet B$	matrix dot-product, i.e., Trace($A^\mathsf{T} B$)
$z(C, s, A, b)$	optimal objective of MESP with parameter C, s, A, b
C	covariance matrix of MESP
s	size of chosen subset in MESP
A, b	parameters of linear constraints $Ax \leq b$
n	size of ground set in MESP
$\|S\|$	cardinality of set S
$S(x)$	support of x
$C[S(x), S(x)]$	principal submatrix of C with rows and columns indexed by $S(x)$
$\{0,1\}^n$	set of length-n binary vectors
$[0,1]^n$	set of length-n vectors with elements between 0 and 1
$\mathbf{e}$	vector of all-ones
$\mathbf{e}_i$	i-th standard unit vector in $\mathbb{R}^n$
x^T	transpose of a vector x
x_i	i-th element of x
x_S	subvector of x indexed by S
$\sqrt{x}$	vector whose elements are square of the corresponding elements of x
x^p	$(x_1^{p_1}, x_2^{p_2}, \ldots, x_n^{p_n})$, x, p are two n-vectors
Diag(x)	diagonal matrix having diagonal elements from x
diag(X)	vector obtained from the diagonal elements of matrix X
$\mathbb{R}^{m \times n}$	set of real matrices of size $m \times n$
$\mathbb{R}^n$	set of real vectors of size n
$\mathbb{R}^n_+$ ($\mathbb{R}^n_{++}$)	set of vector with nonnegative (positive) elements
$\mathbb{S}^n_+$ ($\mathbb{S}^n_{++}$)	set of positive-semidefinite (definite) symmetric matrices
$A \succ B$	$A - B$ is positive-definite
$A \succeq B$	$A - B$ is positive-semidefinite

Table 1.1: Summary of notations used in this dissertation.

Notation	Description
γ	scaling parameter in the scaling technique
ψ	natural logarithm of γ
Υ	general scaling parameter in the general scaling technique
M	mask in the masking technique, i.e., a correlation matrix
$\mathcal{M}_n$	set of masks of order n
J_n / E_n	all-ones matrix of order n
I_n	identity matrix of order n the only nonzero component
E_{ij}^n	order-n square matrix with being a one in the (i, j) position
$\lambda_\ell(A)$	ℓ-th greatest eigenvalue of a matrix $A \in \mathbb{S}_+^n$
$A_{i\cdot}$	i-th row of A
$A_{\cdot j}$	j-th column of A
$A^\dagger$	Moore-Penrose (generalized) inverse of A
$\|\cdot\|$	2-norm of a vector or a matrix
$\mathrm{dom}(f)$	domain of a function f
$\partial f(\cdot)$	subdifferential of f
$\partial f(\cdot; d)$	directional derivative of f in the direction d

Table 1.2: Summary of notations used in this dissertation.

1.5 Dissertation organization

In this work, we will not only establish new upper bounds for MESP and CMESP, but we also expand existing methodologies and introduce new techniques to improve these bounds. Furthermore, we develop rigorous theories and efficient algorithms to determine optimal or near-optimal parameters for the implementation of these techniques, which can also be extended to techniques that may be proposed in the future.

In Chapter 2, we present a methodology for combining different upper bounds of MESP based on convex relaxation to obtain better upper bounds. In Chapter 3, we propose a mild generalization of an upper bound of (Nikolov, 2015) and (Li and Xie, 2023), based on a factorization of an input covariance matrix, which turns out being the state-of-art among many problem instances. In Chapter 4, we extend the scaling technique to *generalized scaling*, employing a positive vector of parameters, which allows much more flexibility to adjust the shape of continuous relaxations of CMESP, and thus significantly reduces the gap between the upper bounds and the optimal value of CMESP further compared to scaling. In Chapter

5, we establish that for many instances, the linx bound can be improved via masking by an amount that is at least linear in the problem size n, even when optimal scaling parameters are employed. We also extend an earlier result that the linx bound is convex in the logarithm of a scaling parameter, making a full characterization of its limiting behavior and providing an efficient means of calculating its limiting behavior in all cases. In chapter 6, we further introduce an advanced limited memory quasi-Newton algorithm to compute an improved mask over the baseline all-ones mask (original bound) for the linx bound, which turns out to achieve state-of-art performance over many benchmark instances.

CHAPTER 2

Mixing Convex-Optimization Bounds

This chapter has been published as:

Zhongzhu Chen, Marcia Fampa, Amélie Lambert, Jon Lee. Mixing convex-optimization bounds for maximum-entropy sampling. *Mathematical Programming*, *Series B*. 188:539-568 (2021). https://doi.org/10.1007/s10107-020-01588-w

2.1 Introduction

Numerous established upper bounds for the optimal value in MESP are derived from convex optimization techniques. This chapter introduces a general methodology for amalgamating these bounds to attain superior ones named "mixing". We provide a universal formula for combining any set of convex optimization-based upper bounds and propose a quasi-Newton method to calculate the optimal combination coefficient. This approach allows the integration of additional improving techniques like scaling, complementing, and masking, to enhance the quality of upper bounds in conjunction with the mixing. Additionally, we present an innovative finding related to the convexity of linx and BQP bounds in the context of the logarithmic scale of the scaling parameter, facilitating the determination of optimal scaling parameters. We implemented this mixing methodology on linx, BQP, and NLP upper bounds using benchmark instances. The empirical data indicates that this mixing strategy substantially improves the upper bounds in numerous instances, particularly when individual bounds exhibit close similarities.

In §2.2, we describe a very simple general idea for "mixing" bounds. In §2.3, we apply the simple idea to MESP by mixing the so-called "BQP bound" (Anstreicher, 2018) with the same bound applied to the complementary problem. In §2.4, we mix the so-called "NLP bound" (Anstreicher, Fampa, Lee, and Williams, 1999) with the same bound applied to the complementary problem. Because the BQP bound and the NLP bound are not invariant under complementation, we can potentially get improved bounds with these mixings. In

§2.5, we look at tuning the so-called "linx bound" (Anstreicher, 2020) and mixing with it. In §2.6, we investigate mixing the NLP bound (or its complement) with the BQP bound (or its complement) — this is not a simple matter because of incompatibility between existing solvers. In §2.7, we present the results of an experiment designed to demonstrate the usefulness of our techniques across a family of instances. In §2.8, we make some concluding remarks.

2.2 General mixing

The idea described here is so simple that we do not dare claim that it is original. We are however confident that it is new in the context of the MESP.

We start with a combinatorial maximization problem

$$z := \max\{f(S) \ : \ S \in \mathcal{F}\},$$

where $\mathcal{F}$ is an arbitrary subset of the power set of $\{1, 2, \ldots, n\}$.

We consider upper bounds for z based on convex relaxation in a possibly lifted space of variables. The compact and convex set $\mathcal{P}_i$ uses variables $(x, \mathcal{X}^i)$ (ignore the i for now; we will use it later). The vector $x \in [0,1]^n$ relaxes $x \in \{0,1\}^n$ and is used to model $\mathcal{F}$. Specifically, we assume that if we project $\mathcal{P}_i$ onto the x coordinates, we get a subset of $[0,1]^n$, and then if we intersect with $\mathbb{Z}^n$, we get precisely the characteristic vectors of $\mathcal{F}$.

Next, we have a concave function f_i, possibly depending on a parameter (vector) ψ_i, taking $(x, \mathcal{X}^i) \in \mathcal{P}_i$ to $\mathbb{R}$. We assume that for $(x, \mathcal{X}^i) \in \mathcal{P}_i$ such that $x \in \{0,1\}^n$, we have $f_i(\psi_i; x, \mathcal{X}^i) = f(S(x))$. In this sense, f_i on $\mathcal{P}_i$ is an *exact relaxation* (possibly in an extended space) of f on $\mathcal{F}$.

Now, we consider having $m \geq 2$ relaxations, indexed by $i = 1, \ldots, m$. So, for $i = 1, 2, \ldots, m$, we have the convex programs

$$v_i(\psi_i) := \max\left\{f_i(\psi_i; x, \mathcal{X}^i) \ : \ (x, \mathcal{X}^i) \in \mathcal{P}_i\right\},$$

yielding m upper bounds on z. Associated with each of these bounds $v_i(\psi_i)$ is the convex relaxation $\max\{f_i(\psi_i; x, \mathcal{X}^i) \ : \ (x, \mathcal{X}^i) \in \mathcal{P}_i\}$.

Next, for $\alpha \in \mathbb{R}^m_+$, such that $\mathbf{e}^\mathsf{T}\alpha = 1$, and $\psi := (\psi_1^\mathsf{T}, \ldots, \psi_m^\mathsf{T})^\mathsf{T}$, we define the *mixing bound*

$$v(\alpha, \psi) := \max\left\{\sum_{i=1}^m \alpha_i f_i(\psi_i; x, \mathcal{X}^i) \ : \ (x, \mathcal{X}^i) \in \mathcal{P}_i \ , \ 1 = 1, 2, \ldots, m\right\}. \tag{2.1}$$

It is a natural goal to optimize the mixing bound over both of the parameters α and ψ.

But generally this is not tractable. We will soon see that generally we can optimize on α, and in some situations we can optimize on ψ.

The following is very simple to establish.

Proposition 2.1. *For fixed ψ, the function $v(\alpha, \psi)$ is convex on $\{\alpha \in \mathbb{R}^m \ : \ \alpha \geq 0\}$, and for all $\alpha \in \mathbb{R}^m$ such that $\mathbf{e}^\mathsf{T}\alpha = 1$, we have $v(\alpha, \psi) \geq z$.*

Owing to this, a natural goal is to seek the best mixing bound by solving the convex problem

$$\min_\alpha \left\{ v(\alpha, \psi) \ : \ \mathbf{e}^\mathsf{T}\alpha = 1, \ \alpha \in \mathbb{R}^m_+ \right\} \ . \tag{2.2}$$

The power of the mixing bound is that the same variable x is appearing in each of the $\mathcal{P}_i$. If it were not for this, then the minimum value in (2.2) would trivially be $\min_{i=1}^m v_i(\psi_i)$.

Of course each $\mathcal{P}_i$ can be strengthened to improve the mixing bound. But very importantly, we note that the mixing bound can be strengthened by introducing valid equations and inequalities across the entire variable space: $x, \mathcal{X}_1, \ldots, \mathcal{X}_m$. We exploit both of these observations in the next section.

Before continuing, we wish to mention that a slightly different formulation for finding an optimal mixing is as the following convex program.

$$\begin{aligned}
&\max \ v \\
&\text{subject to:} \\
&v \leq f_i(\psi_i; x, \mathcal{X}^i), \ i = 1, 2, \ldots, m; \\
&(x, \mathcal{X}^i) \in \mathcal{P}_i \ , \ 1 = 1, 2, \ldots, m.
\end{aligned}$$

The equivalence can easily be seen by Lagrangian duality. We prefer our formulation because by aggregating the nonlinearities into the objective, in the style of a surrogate dual, we get a formulation that is more easily handled by solvers and more easily optimized in terms of selecting good mixing (and other bound) parameters. Related to this, in the context of branch-and-bound, we can expect that child subproblems will be able to inherit good parameters from their parents, leading to faster computations.

Next, we discuss strategies for solving (2.2). Let

$$(x^*, \mathcal{X}^*) := (x^*, \mathcal{X}^{1*}, \ldots, \mathcal{X}^{m*}) := (x^*(\alpha, \psi), \mathcal{X}^{1*}(\alpha, \psi), \ldots, \mathcal{X}^{m*}(\alpha, \psi))$$

be an optimal solution of (2.1), for given α and ψ, and let $f_i^*(\alpha, \psi) := f_i(\psi_i; x^*, \mathcal{X}^{i*})$,

$i = 1, \ldots, m$. Then,

$$v(\alpha, \psi) = \sum_{i=1}^{m} \alpha_i f_i^*(\alpha, \psi) \ .$$

We have the following simple but useful result.

Proposition 2.2. *For fixed ψ,*

$$g_\alpha(\alpha, \psi) := (f_1^*(\alpha, \psi), f_2^*(\alpha, \psi), \ldots, f_m^*(\alpha, \psi))^\mathsf{T}$$

is a subgradient of v (with respect to α) at $\alpha \in \mathbb{R}^m_+$ such that $\mathbf{e}^\mathsf{T}\alpha = 1$.

Proof. For $\tilde{\alpha} \in \mathbb{R}^m$ such that $\mathbf{e}^\mathsf{T}\tilde{\alpha} = 1$,

$$\begin{aligned} v(\alpha, \psi) + (\tilde{\alpha} - \alpha)^\mathsf{T} g_\alpha(\alpha, \psi) &= \sum_{i=1}^{m} \alpha_i f_i^*(\alpha, \psi) + \sum_{i=1}^{m} \tilde{\alpha}_i f_i^*(\alpha, \psi) - \sum_{i=1}^{m} \alpha_i f_i^*(\alpha, \psi) \\ &= \sum_{i=1}^{m} \tilde{\alpha}_i f_i^*(\alpha, \psi) \leq \sum_{i=1}^{m} \tilde{\alpha}_i f_i^*(\tilde{\alpha}, \psi) = v(\tilde{\alpha}, \psi). \end{aligned}$$

where the inequality holds because $(x^*, \mathcal{X}^*)$ need not be optimal for $\tilde{\alpha}$. □

Via the standard projected subgradient algorithm (optimizing $v(\alpha, \psi)$ over the simplex $\{\alpha \in \mathbb{R}^m_+ \ : \ \mathbf{e}^\mathsf{T}\alpha = 1\}$), this already gives us a convergent algorithm for (2.2). Additionally, when $m = 2$, we can rewrite v as a function of α_1 alone ($\alpha_2 = 1 - \alpha_1$), and then the subgradient for the now univariate v becomes the scalar $f_1^*(\alpha_1, \psi) - f_2^*(\alpha_1, \psi)$. With this, we can use a simple bisection search for $\alpha_1 \in [0, 1]$, considering the sign of $f_1^*(\alpha_1, \psi) - f_2^*(\alpha_1, \psi)$ at each iteration, to get an improved algorithm when $m = 2$.

However, we have found that even for $m = 2$, we can get better practical convergence with an interior point algorithm aimed at (2.2). We solve (2.2) with a logarithmic barrier method, considering

$$\min \left\{ v(\alpha, \psi) - \mu \sum_{i=1}^{m} \log(\alpha_i) \ : \ \mathbf{e}^\mathsf{T}\alpha = 1, \ \alpha > 0, \ \alpha \in \mathbb{R}^m \right\}, \tag{2.3}$$

where $\mu > 0$ is the barrier parameter.

We note that v is not everywhere differentiable, but it is at α for which (2.1) has a unique optimum (Fiacco, 1983, Corollary 3.4.2) or (Fiacco and Ishizuka, 1990a, Theorem 4.1), in which case the subgradient identified is the gradient. In the implementation of the barrier method, we use the subgradient g_α from Proposition 2.2 to replace the gradient of v. We then approximate the Hessian of v by a positive definite matrix B, which is initialized

as the identity matrix and updated at each iteration with a BFGS approach. Working in the affine set $\{\alpha \in \mathbb{R}^m \ : \ \mathbf{e}^\mathsf{T}\alpha = 1\}$, we can apply the classical result that under sufficient smoothness, BFGS converges with weak Wolfe line searches for unconstrained convex minimization (Powell, 1976). In this way, we get a convergent algorithm (under smoothness assumptions)[1], for any fixed barrier parameter. If the barrier parameter is slowly reduced, then in theory we converge to the solution of (2.1) (Fiacco and McCormick, 1968).

For practical purposes, we work a bit differently. In Algorithm 1, we present, in detail, one inner iteration of our barrier method. At an inner iteration, μ is fixed, and we are carrying out one iteration of a quasi-Newton method, working toward the solution of the barrier problem (2.3). The search direction of the barrier method is given by the projection of the quasi-Newton direction computed over B and g_α, onto the null space of $\mathbf{e}^\mathsf{T}$. To compute the direction at each iteration, we need to compute $f_i^*(\alpha, \psi)$, for $i = 1, \ldots, m$, and therefore we need the optimal solution $(x^*, \mathcal{X}^*)$ of (2.1), for the current α. Rather than doing a Wolfe line search, we take a constant fraction τ of the full step if we can, otherwise a constant fraction of the distance to the boundary. The (inner) iteration presented is repeated for a fixed value of the barrier parameter μ, for a prescribed number of times or until the norm of the residual r is small enough. The parameter μ is then reduced and the process repeated, until μ is also small enough.

Next we consider situations in which $f_i(\psi_i; x, \mathcal{X}^i)$ is convex in ψ_i, for each fixed $(x, \mathcal{X}^i)$, $i = 1, \ldots, m$, and we wish to optimize the bound $v(\alpha, \psi)$ over the parameter ψ, for fixed $\alpha \in \mathbb{R}^m_+$, such that $\mathbf{e}^\mathsf{T}\alpha = 1$:

$$\min_\psi \{v(\alpha, \psi)\} \ . \tag{2.4}$$

Proposition 2.3. *For fixed $\alpha \in \mathbb{R}^m_+$, such that $\mathbf{e}^\mathsf{T}\alpha = 1$, let g_{ψ_i} be a subgradient of $f_i(\psi_i; x^*, \mathcal{X}^{i*})$ with respect to ψ_i. Then*

$$g_\psi(\alpha, \psi) := \left(\alpha_1 g_{\psi_1}^\mathsf{T}, \ldots, \alpha_m g_{\psi_m}^\mathsf{T}\right)^\mathsf{T}$$

is a subgradient of v (with respect to ψ) at ψ.

Proof. For arbitrary $\tilde{\psi}$,

$$v(\alpha, \psi) + (\tilde{\psi} - \psi)^\mathsf{T} g_\psi(\alpha, \psi) = \sum_{i=1}^m \alpha_i f_i^*(\alpha, \psi) + \sum_{i=1}^m (\tilde{\psi}_i - \psi_i)^\mathsf{T} \alpha_i g_{\psi_i}$$

[1]We note that even in the nondifferentiable case, quasi-Newton methods have very good convergence properties (Lewis and Overton, 2013).

Algorithm 1: Updating α

Input: k, α^k, $g_\alpha(\alpha^k,\psi) = (f_1^*(\alpha^k,\psi), f_2^*(\alpha^k,\psi), \ldots, f_m^*(\alpha^k,\psi))^\mathsf{T}$, B_k.
Compute the residual:

$$r_i := f_i^*(\alpha^k,\psi) - \frac{\mu}{\alpha_i^k}, \quad i = 1,\ldots,m.$$

Solve for δ_α:

$$B_\mu \delta_\alpha = -r,$$

where

$$B_\mu := B_k + \mu\,\mathrm{Diag}(\alpha^k)^{-2}.$$

Let $\hat{\delta}_\alpha$ be the projection of δ_α onto the null space of $\mathbf{e}^\mathsf{T}$.
Update α:

$$\alpha^{k+1} := \alpha^k + \hat{\theta}\hat{\delta}_\alpha,$$

where

$$\hat{\theta} := \tau \times \min\{1, \mathrm{argmax}_\theta\{\alpha^k + \theta\hat{\delta}_\alpha \geq 0\}\}.$$

Obtain the optimal solution $(x^*, \mathcal{X}^*)$ of (2.1), considering $\alpha = \alpha^{k+1}$, and let

$$f_i^*(\alpha^{k+1},\psi) := f_i(\psi_i; x^*, \mathcal{X}^{i*}), \quad i = 1,\ldots,m.$$

Compute:

$$\begin{aligned} &g_\alpha(\alpha^{k+1},\psi) := (f_1^*(\alpha^{k+1},\psi), f_2^*(\alpha^{k+1},\psi), \ldots, f_m^*(\alpha^{k+1},\psi))^\mathsf{T},\\ &y_k := g_\alpha(\alpha^{k+1},\psi) - g_\alpha(\alpha^k,\psi),\\ &s_k := \alpha^{k+1} - \alpha^k. \end{aligned}$$

if $s_k^\mathsf{T} y_k > 0$ **then**

$$B_{k+1} := B_k - \frac{B_k s_k s_k^\mathsf{T} B_k}{s_k^\mathsf{T} B_k s_k} + \frac{y_k y_k^\mathsf{T}}{y_k^\mathsf{T} s_k},$$

else

$$B_{k+1} := B_k.$$

$k := k + 1$.
Output: k, α^k, $g_\alpha(\alpha^k,\psi)$, B_k.

$$\leq \sum_{i=1}^m \alpha_i f_i(\tilde{\psi}_i; x^*, \mathcal{X}^{i*}) \leq \sum_{i=1}^m \alpha_i f_i^*(\alpha,\tilde{\psi}) = v(\alpha,\tilde{\psi}),$$

where the first inequality holds because g_{ψ_i} is a subgradient of $f_i(\psi_i; x^*, \mathcal{X}^{i*})$ (for $i = 1,\ldots,m$), and the second inequality holds because $(x^*, \mathcal{X}^*)$ need not be optimal for $\tilde{\psi}$. $\square$

We solve (2.4) by a quasi-Newton method. We note that v is not everywhere differentiable, but when the f_i are continuously differentiable, v is differentiable at ψ for which (2.1) has a unique solution; in which case the subgradient referred to in Proposition 2.3 is the gradient. Similarly to what we do in Algorithm 1, we approximate the Hessian by a positive definite matrix B, which is initialized as the identity matrix and updated at each iteration with a BFGS approach. In Algorithm 2, we present an iteration of the quasi-Newton method. The iteration presented is repeated for a prescribed number of times or until the absolute value of the residuals, components of $g_\psi(\alpha, \psi)$, are small enough.

Remark. *We have been assuming that there are no constraints on the parameters ψ_i, because that is the case in our applications to MESP. But if for example each ψ_i were constrained to be in a polyhedron, we could adapt Algorithm 2 with barrier terms, in the manner of Algorithm 1.*

Remark. *When* (2.1) *has a* unique *optimal solution, the gradients g_α and g_ψ asked for by Algorithms 1 and 2, respectively, are correctly identified. Though, as we have mention, BFGS has nice convergence properties even in nonsmooth situations (Lewis and Overton, 2013). In any case, it is natural to ask whether the 'unique-optimum property' holds for given mixing bounds — it is easy to construct artificial situations where it does not hold. A simple and often checkable sufficient condition is that each of the bounds to be mixed has a strictly concave objective function, which implies the same for the mixed bound, for $\alpha \geq 0$, $\mathbf{e}^\mathsf{T}\alpha = 1$. We will consider this in the next sections, as we investigate various mixing bounds for MESP.*

2.3 Mixing the BQP bound with the complementary BQP bound

In this section, we apply the simple mixing idea from §2.2, mixing the (scaled) BQP bound for MESP (Anstreicher, 2018) with the same bound applied to the complementary problem. We will see that minimizing this bound over α gives us a bound that is sometimes stronger than the two bounds that it is based upon — it is always at least as strong. In fact, we will see that the bound will tend to be stronger when the two bounds being mixed have similar values.

Algorithm 2: Updating ψ

Input: k, ψ^k, $g_\psi(\alpha, \psi^k) = (\alpha_1 g_{\psi_1}^\mathsf{T}, \ldots, \alpha_m g_{\psi_m}^\mathsf{T})^\mathsf{T}$, B_k.
Compute the residual:

$$r_i := g_{\psi_i}, \quad i = 1, \ldots, m,$$
$$r := (r_1^\mathsf{T}, r_2^\mathsf{T}, \ldots, r_m^\mathsf{T})^\mathsf{T}.$$

Solve for δ_ψ:

$$B_k \delta_\psi = -r.$$

Update ψ:

$$\psi^{k+1} := \psi^k + \delta_\psi.$$

Obtain the optimal solution $(x^*, \mathcal{X}^*)$ of (2.1) considering $\psi = \psi^{k+1}$, and let g_{ψ_i} be a subgradient of $f_i(\gamma_i; x^*, \mathcal{X}^{i*})$ with respect to ψ_i, $i = 1, \ldots, m$. Normally:

$$g_{\psi_i} := \nabla_{\psi_i} f_i(\psi_i^{k+1}; x^*, \mathcal{X}^{i*}), \quad i = 1, \ldots, m.$$

Compute:

$$g_\psi(\alpha, \psi^{k+1}) := (\alpha_1 g_{\psi_i}^\mathsf{T}, \ldots, \alpha_m g_{\psi_m}^\mathsf{T})^\mathsf{T},$$
$$y_k := g_\psi(\alpha, \psi^{k+1}) - g_\psi(\alpha, \psi^k),$$
$$s_k := \psi^{k+1} - \psi^k.$$

if $s_k^\mathsf{T} y_k > 0$ **then**

$$B_{k+1} := B_k - \frac{B_k s_k s_k^\mathsf{T} B_k}{s_k^\mathsf{T} B_k s_k} + \frac{y_k y_k^\mathsf{T}}{y_k^\mathsf{T} s_k},$$

else

$$B_{k+1} := B_k.$$

$k := k + 1$.
Output: k, ψ^k, $g_\psi(\alpha, \psi^k)$, B_k.

2.3.1 Mixing BQP and its complement

Let

$$P(n, s) := \{(x, X) \in \mathbb{R}^n \times S^n(\mathbb{R}) \ : \ X - xx^\mathsf{T} \succeq 0, \ \operatorname{diag}(X) = x, \ \mathbf{e}^\mathsf{T} x = s, \ X\mathbf{e} = sx\}$$
$$Q(n, n-s) := \{(y, Y) \in \mathbb{R}^n \times S^n(\mathbb{R}) \ :$$

$$Y - yy^\mathsf{T} \succeq 0,\ \operatorname{diag}(Y) = y,\ \mathbf{e}^\mathsf{T} y = n - s,\ Y\mathbf{e} = (n-s)y\}.$$

The set $P(n,s)$ (respectively, $Q(n, n-s)$) is the well-known SDP relaxation of the binary solutions to $X - xx^\mathsf{T} = 0$, $\mathbf{e}^\mathsf{T} x = s$ (respectively, $Y - yy^\mathsf{T} = 0$, $\mathbf{e}^\mathsf{T} y = n - s$).

We introduce the *mixed BQP (mBQP) bound*:

$$\begin{aligned}
&v(C, s; \alpha, \gamma_1, \gamma_2) := \\
&\max\ (1-\alpha)\left(\operatorname{ldet}(\gamma_1 C \circ X + \operatorname{Diag}(\mathbf{e} - x)) - s\log\gamma_1\right) \\
&\quad + \alpha\left(\operatorname{ldet}\left(\gamma_2 C^{-1} \circ Y + \operatorname{Diag}(\mathbf{e} - y)\right) - (n-s)\log\gamma_2 + \operatorname{ldet} C\right), \\
&\text{subject to:} \\
&\quad (x, X) \in P(n,s),\ (y, Y) \in Q(n, n-s),\ x + y = \mathbf{e},
\end{aligned}$$

where $0 \le \alpha \le 1$ is a "weighting" parameter, and $\gamma_1, \gamma_2 > 0$ are "scaling parameters" (Anstreicher, 2018) for the first treatment of scaling for a BQP bound). We will see that this mBQP bound is a manifestation of the idea from §2.2, mixing the scaled BQP bound with its complement.

It is almost immediate that the mBQP bound is a mixing in the precise sense of §2.2, but because of the way that we have formulated it with different variables for the complementary part, there is a little checking to do.

We define an invertible linear map Φ by

$$\Phi(x, X) = (\mathbf{e} - x, X + \mathbf{e}\mathbf{e}^\mathsf{T} - \mathbf{e}x^\mathsf{T} - x\mathbf{e}^\mathsf{T}).$$

Notice that if $(\hat{y}, \hat{Y}) := \Phi(\hat{x}, \hat{X})$, then $\hat{Y}_{ij} = \hat{X}_{ij} + 1 - \hat{x}_j - \hat{x}_i$.

We have the following useful result.

Lemma 2.4. *$(\hat{x}, \hat{X}) \in P(n,s)$ if and only if $\Phi(\hat{x}, \hat{X}) \in Q(n, n-s)$.*

Proof. We check the constraints:

$$\begin{aligned}
&\hat{Y} - \hat{y}\hat{y}^\mathsf{T} = \hat{X} + \mathbf{e}\mathbf{e}^\mathsf{T} - \mathbf{e}\hat{x}^\mathsf{T} - \hat{x}\mathbf{e}^\mathsf{T} - (\mathbf{e} - \hat{x})(\mathbf{e} - \hat{x})^\mathsf{T} = \hat{X} - \hat{x}\hat{x}^\mathsf{T} \succeq 0\ . \\
&\operatorname{diag}(\hat{Y}) = \operatorname{diag}(\hat{X}) + \operatorname{diag}(\mathbf{e}\mathbf{e}^\mathsf{T}) - \operatorname{diag}(\mathbf{e}\hat{x}^\mathsf{T}) - \operatorname{diag}(\hat{x}\mathbf{e}^\mathsf{T}) = \hat{x} + \mathbf{e} - \hat{x} - \hat{x} = \hat{y}\ . \\
&\mathbf{e}^\mathsf{T}\hat{y} = \mathbf{e}^\mathsf{T}(\mathbf{e} - \hat{x}) = n - s\ . \\
&\hat{Y}\mathbf{e} = \left(\hat{X} + \mathbf{e}\mathbf{e}^\mathsf{T} - \mathbf{e}\hat{x}^\mathsf{T} - \hat{x}\mathbf{e}^\mathsf{T}\right)\mathbf{e} = s\hat{x} + n\mathbf{e} - s\mathbf{e} - n\hat{x} = (n-s)(\mathbf{e} - \hat{x}) = (n-s)\hat{y}\ .
\end{aligned}$$

The other direction is similar. □

For $\alpha = 0$ and $\alpha = 1$, the mBQP reduces to the bounds of (Anstreicher, 2018)[2]:

Proposition 2.5. *$v_{BQP}(C, s; \alpha = 0, \gamma_1, \gamma_2)$ is equal to the scaled BQP bound*

$$\begin{aligned} -s\log\gamma_1 \;+\; & \max \;\; \operatorname{ldet}\left(\gamma_1 C \circ X + \operatorname{Diag}(\mathbf{e} - x)\right) \;, \\ & \textit{subject to:} \\ & (x, X) \in P(n, s), \end{aligned}$$

and $v_{BQP}(C, s; \alpha = 1, \gamma_1, \gamma_2)$ is equal to the scaled complementary BQP bound

$$\begin{aligned} \operatorname{ldet} C - (n-s)\log\gamma_2 \;+\; & \max \;\; \operatorname{ldet}\left(\gamma_2 C^{-1} \circ Y + \operatorname{Diag}(\mathbf{e} - y)\right) \\ & \textit{subject to:} \\ & (y, Y) \in Q(n, n-s). \end{aligned}$$

Proof. When $\alpha = 0$, for any $(\hat{x}, \hat{X}) \in P(n, s)$, Lemma 2.4 allows us to always be able to choose a $(\hat{y}, \hat{Y})$, which together with $(\hat{x}, \hat{X})$ is feasible for the mBQP optimization formulation. And because $\alpha = 0$, the choice of $(\hat{y}, \hat{Y})$ has no impact on the mBQP objective function. Similarly, when $\alpha = 1$, for any $(\hat{y}, \hat{Y}) \in Q(n, n-s)$, Lemma 2.4 allows us to always be able to choose a $(\hat{x}, \hat{X})$ which together with $(\hat{y}, \hat{Y})$ is feasible for the mBQP optimization formulation. And because $\alpha = 1$, the choice of $(\hat{x}, \hat{X})$ has no impact on the mBQP objective function. □

Of course we have

Proposition 2.6. *For all $\gamma_1 > 0$, $\gamma_2 > 0$, $0 \leq \alpha \leq 1$,*

$$z(C, s) \leq v(C, s; \alpha, \gamma_1, \gamma_2) \;.$$

We can see from the convexity of v that there is a good potential to improve on the minimum of the scaled BQP bound and the scaled complementary BQP bound precisely when these two bounds are similar. See Figure 2.1 where this is illustrated using the "$n = 63$" benchmark covariance matrix from the literature. This matrix (and an "$n = 124$" one that we use later), obtained from J. Zidek (University of British Columbia), coming from an application to re-designing an environmental monitoring network, has been used extensively in testing and developing algorithms for MESP; see (Ko, Lee, and Queyranne, 1995; Lee, 1998; Anstreicher, Fampa, Lee, and Williams, 1999; Lee and Williams, 2003; Hoffman, Lee,

[2]Helmberg suggested (essentially) the BQP bound in 1995 (Lee, 2012; Fedorov and Lee, 2000) to Anstreicher and Lee, but no one developed it at all until (Anstreicher, 2018) did so extensively, drawing in and significantly extending some techniques from (Anstreicher, Fampa, Lee, and Williams, 1999).

and Williams, 2001; Anstreicher and Lee, 2004; Burer and Lee, 2007; Anstreicher, 2018, 2020). The case of $s = 10$ shows the improvement from mixing when the two individual bounds are approximately equal. The case of $s = 16$ shows that mixing can give some improvement even when one bound is substantially higher than the other. A good value for α can be found using a univariate search or by applying the logarithmic-barrier algorithm to (2.3), using Algorithm 1 to update α. Moreover, in the context of branch-and-bound for exact solution of the MESP, a good (starting) value of α can be inherited from a parent.

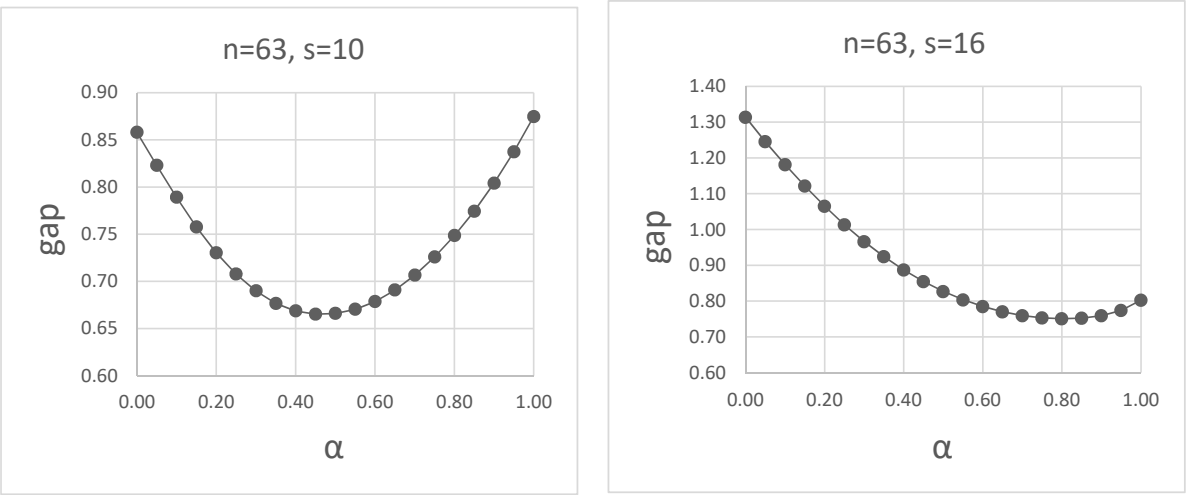

Figure 2.1: Gap vs. α (optimized γ_i)

2.3.2 Valid equations in the extended spaces

Next, we will see that we can strengthen the mBQP bound, using equations that link the extended variables from the two bounds that we mix, and then even eliminate the variables (y, Y).

Proposition 2.7.

$$
\begin{aligned}
& v(C, s; \alpha, \gamma_1, \gamma_2) \;\geq\; \check{v}(C, s; \alpha, \gamma_1, \gamma_2) := \\
& \max \;\; (1-\alpha)\left(\operatorname{ldet}\left(\gamma_1 C \circ X + \operatorname{Diag}(\mathbf{e} - x)\right) - s\log\gamma_1\right) \\
& \quad + \alpha\left(\operatorname{ldet}\left(\gamma_2 C^{-1} \circ (X + \mathbf{e}\mathbf{e}^{\mathsf{T}} - \mathbf{e}x^{\mathsf{T}} - x\mathbf{e}^{\mathsf{T}}) + \operatorname{Diag}(x)\right)\right. \\
& \qquad \left. -(n-s)\log\gamma_2 + \operatorname{ldet} C\right), \\
& \textit{subject to:}
\end{aligned}
$$

$$(x, X) \in P(n, s).$$

The result follows from Lemma 2.4 and the following simple lemma.

Lemma 2.8. *For the solutions of $x + y = e$, $X = xx^\mathsf{T}$, $Y = yy^\mathsf{T}$, the equations $Y = X + \mathbf{e}\mathbf{e}^\mathsf{T} - \mathbf{e}x^\mathsf{T} - x\mathbf{e}^\mathsf{T}$ are valid.*

Proof. Under $x + y = \mathbf{e}$, we have that

$$0 = Y - yy^\mathsf{T} = Y - (\mathbf{e} - x)(\mathbf{e} - x)^\mathsf{T} = Y - \mathbf{e}\mathbf{e}^\mathsf{T} + \mathbf{e}x^\mathsf{T} + x\mathbf{e}^\mathsf{T} - xx^\mathsf{T} \ .$$

Subtracting $0 = X - xx^\mathsf{T}$, we obtain the desired equations. □

We experimented further with the "$n = 63$" covariance matrix. Considering now Figure 2.2, the unmixed bounds are indicated by the lines for "$\alpha = 0$" and "$\alpha = 1$". We first optimized the γ_i for these bounds (see §2.3.3). We chose an interesting range of s, where the unmixed bounds transition between which is stronger (i.e., the lines cross). The line indicated by "α^*" is the optimal mixing of the BQP bound and its complement. Note that we only optimized $v(C, s; \alpha, \gamma_1, \gamma_2)$ on α, keeping the optimal γ_i from the unmixed bounds. A (probably small) further improvement could be obtained by iterating between optimizing on α and the γ_i; this is considered in detail in §2.3.3.2. The line indicated by "α^* strengthened" is the optimal mixing of the BQP bound and its complement, but now with the valid equations in the extended space. Note that again we only optimized $\check{v}(C, s; \alpha, \gamma_1, \gamma_2)$ on α, keeping the optimal γ_i from the unmixed bounds.

We can also seek to improve the mBQP bound by adding RLT, triangle and other inequalities, valid for the BQP, for both (x, X) and (y, Y). We could do this directly (like (Anstreicher, 2018)), but the conic-bundle method (Fischer, Gruber, Rendl, and Sotirov, 2006) seems more promising, due to the large number of inequalities to be potentially exploited. So we dynamically include triangle inequalities via a bundle method; specifically we use the solver SDPT3 (see (Toh, Todd, and Tütüncü, 1999)) together with the Conic Bundle Library (Helmberg, 2005–2019) for solving the associated semidefinite programs, as described in (Billionnet, Elloumi, Lambert, and Wiegele, 2017). In the figure, the line "α^* strengthened + triangles" indicates the bound obtained.

We repeated this experiment for the larger "$n = 124$" benchmark covariance matrix from the literature. The results, exhibiting a similar behavior, are indicated in Figure 2.3.

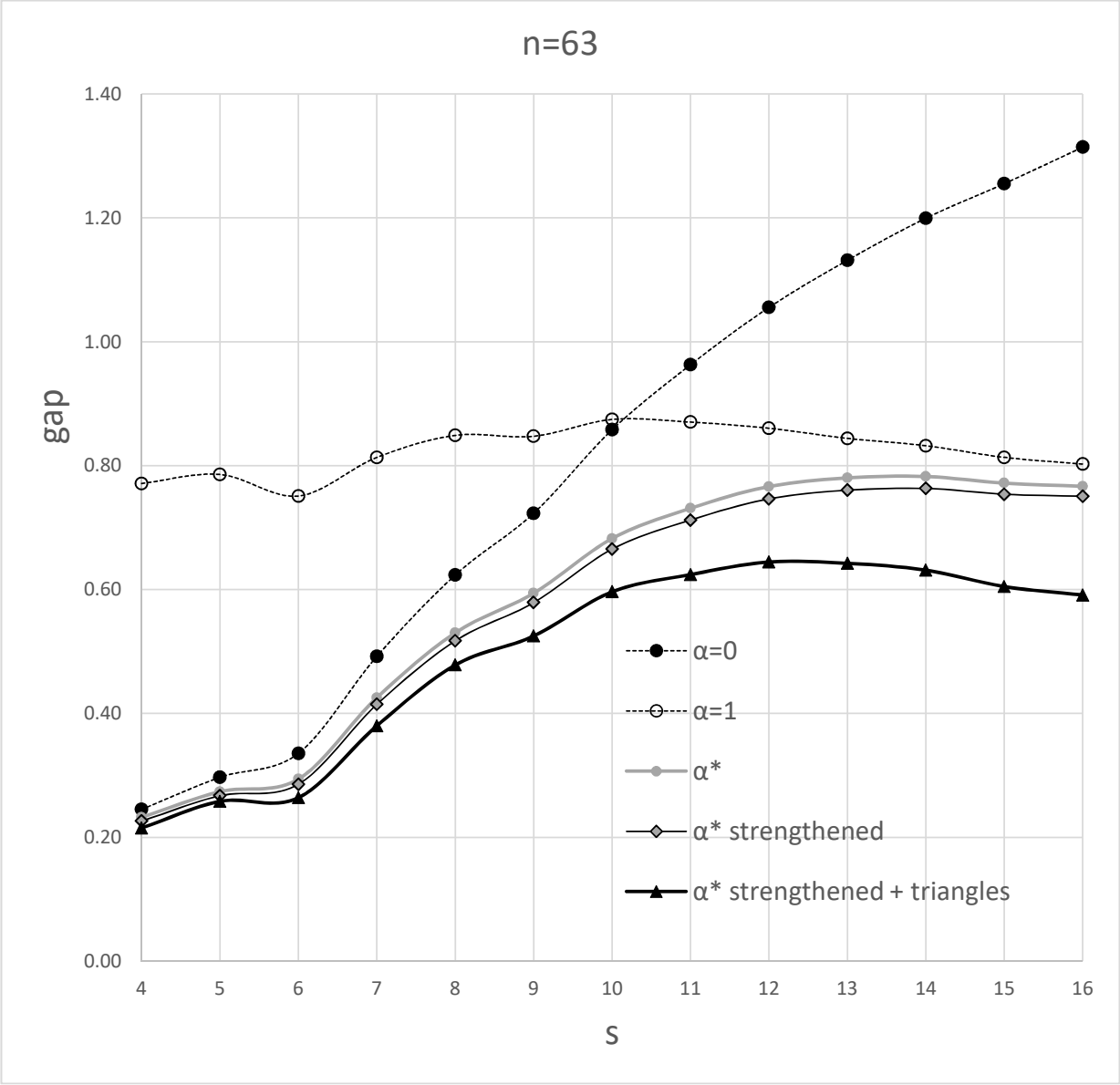

Figure 2.2: Gap vs. s (optimized α and γ_i)

2.3.3 Choosing good parameters $(\alpha, \gamma_1, \gamma_2)$

Toward designing a reasonable algorithm for jointly minimizing $\check{v}(C, s; \alpha, \gamma_1, \gamma_2)$, over $\alpha \in [0, 1]$ and $\gamma_1, \gamma_2 > 0$, we establish convexity properties.

2.3.3.1 Convexity properties

Theorem 2.9. *For fixed $\gamma_1, \gamma_2 > 0$, the function $\check{v}(C, s; \alpha, \gamma_1, \gamma_2)$ is convex in $\alpha \in [0, 1]$. For fixed $\alpha \in [0, 1]$, the function $\check{v}(C, s; \alpha, \exp(\psi_1), \exp(\psi_2))$ is jointly convex in $(\psi_1, \psi_2) \in \mathbb{R}^2$.*

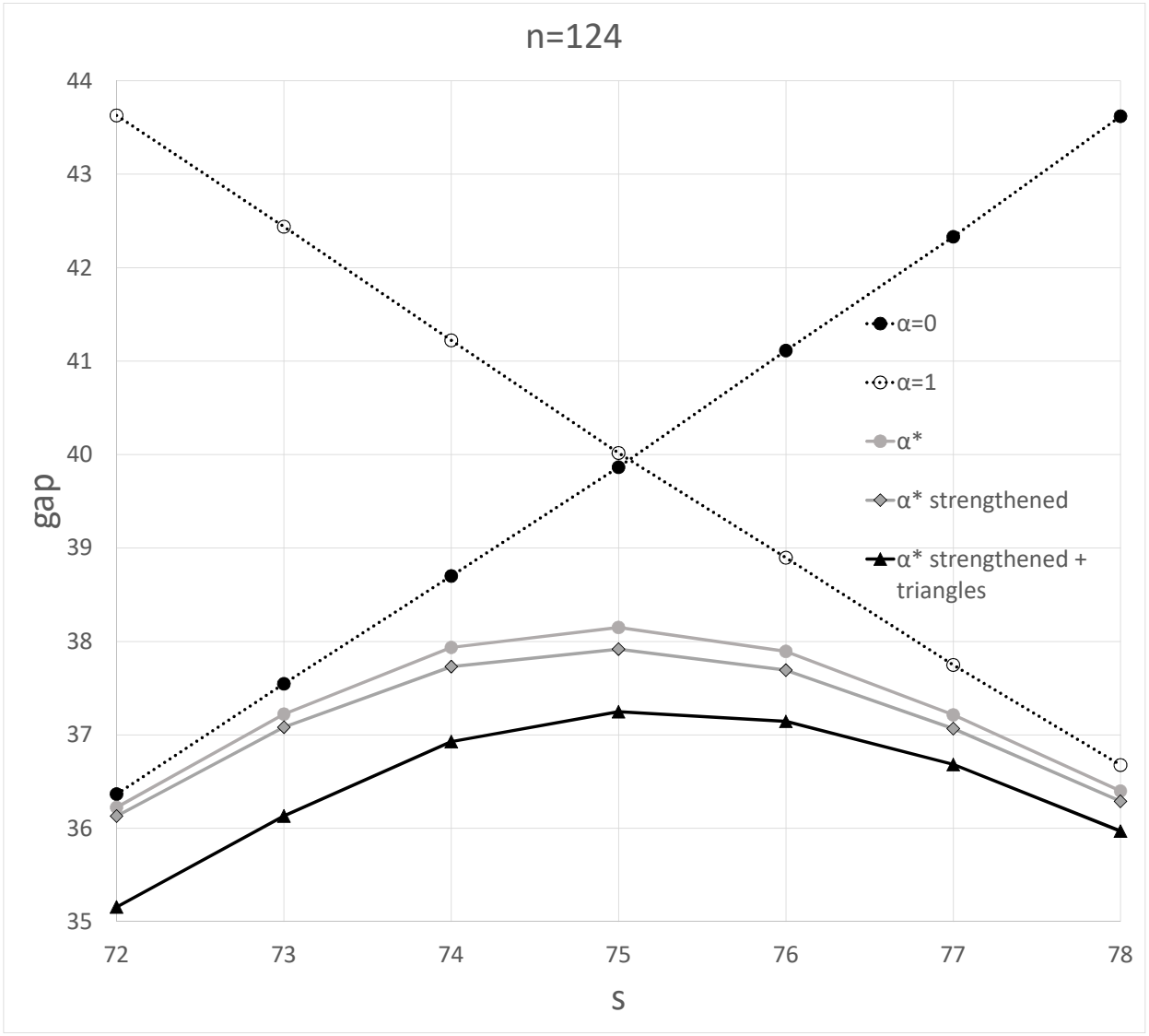

Figure 2.3: Gap vs. s (optimized α and γ_i)

Proof. We already know from general principles that our mixing bounds are convex in α. So in this section, we begin by establishing joint convexity in the logarithms of the scaling parameters γ_1, γ_2.

Let

$$F_1(C, s; \gamma_1; x, X) := (\gamma_1 C - I) \circ X + I = \gamma_1 C \circ X + \mathrm{Diag}(\mathbf{e} - x), \tag{2.5}$$

$$F_2(C, s; \gamma_2; x, X) := \gamma_2 C^{-1} \circ (X + \mathbf{e}\mathbf{e}^{\mathsf{T}} - \mathbf{e}x^{\mathsf{T}} - x\mathbf{e}^{\mathsf{T}}) + \mathrm{Diag}(x), \tag{2.6}$$

$$f_1(C, s; \gamma_1; x, X) := \mathrm{ldet}\, F_1(C, s; \gamma_1; x, X) - s \log \gamma_1,$$

$$f_2(C, s; \gamma_2; x, X) := \operatorname{ldet} F_2(C, s; \gamma_2; x, X) - (n - s) \log \gamma_2 + \operatorname{ldet} C,$$

$$f(C, s; \alpha, \gamma_1, \gamma_2; x, X) := (1 - \alpha) f_1(C, s; \gamma_1; x, X) + \alpha f_2(C, s; \gamma_2; x, X).$$

So, with this notation,

$$\check{v}(C, s; \alpha, \gamma_1, \gamma_2) = \max_{(x,X) \in P(n,s)} (1 - \alpha) f_1(C, s; \gamma_1; x, X) + \alpha f_2(C, s; \gamma_2; x, X).$$

The function $\check{v}(C, s; \alpha, \exp(\psi_1), \exp(\psi_2))$ is the point-wise maximum of $f(C, s; \alpha$, $\exp(\psi_1), \exp(\psi_2); x, X)$, over $(x, X) \in P(n, s)$. So it suffices to show (Boyd and Vandenberghe, 2004, top of p. 81, §3.2.3) that $f(C, s; \alpha, \exp(\psi_1), \exp(\psi_2); x, X)$ is itself convex for each fixed $(x, X) \in P(n, s)$.

In what follows, for $i = 1, 2$, we use f_i and $f_i(\gamma_i; x, X)$ as short forms for $f_i(C, s; \gamma_i; x, X)$, and we use $F_i(\gamma_i; x, X)$ as a short form for $F_i(C, s; \gamma_i; x, X)$. We have

$$\begin{aligned}
\frac{\partial f_1}{\partial \gamma_1} &= \frac{\partial}{\partial \gamma_1} \left(\operatorname{ldet} F_1(\gamma_1; x, X) - s \log \gamma_1 \right) \\
&= \frac{\partial}{\partial \gamma_1} \left(\operatorname{ldet}(\gamma_1 C \circ X + \operatorname{Diag}(\mathbf{e} - x)) - s \log \gamma_1 \right) \\
&= F_1(\gamma_1, ; x, X)^{-1} \bullet (C \circ X) - \frac{s}{\gamma_1} \\
&= \frac{1}{\gamma_1} \left(F_1(\gamma_1; x, X)^{-1} \bullet (\gamma_1 C \circ X) - s \right) \\
&= \frac{1}{\gamma_1} \left(F_1(\gamma_1; x, X)^{-1} \bullet F_1(\gamma_1; x, X) - F_1(\gamma_1; x, X)^{-1} \bullet \operatorname{Diag}(\mathbf{e} - x) - s \right) \\
&= \frac{1}{\gamma_1} \left(n - s - F_1(\gamma_1; x, X)^{-1} \bullet \operatorname{Diag}(\mathbf{e} - x) \right) \ .
\end{aligned}$$

Letting $\psi_1 := \log \gamma_1$, by the chain rule we have

$$\frac{\partial f_1}{\partial \gamma_1} = \frac{\partial f_1}{\partial \psi_1} \frac{d\psi_1}{d\gamma_1} = \frac{\partial f_1}{\partial \psi_1} \frac{1}{\gamma_1} \ .$$

So we have

$$\frac{\partial f_1}{\partial \psi_1} = \gamma_1 \frac{\partial f_1}{\partial \gamma_1} = n - s - F_1(\exp(\psi_1); x, X)^{-1} \bullet \operatorname{Diag}(\mathbf{e} - x) =: \overline{g}_1(\gamma_1) \ .$$

Next, we calculate

$$\frac{\partial^2 f_1}{\partial \gamma_1^2} = \frac{\partial}{\partial \gamma_1} \left(\frac{1}{\gamma_1} \left(n - s - F_1(\gamma_1; x, X)^{-1} \bullet \operatorname{Diag}(\mathbf{e} - x) \right) \right)$$

$$= -\frac{1}{\gamma_1^2}\left(n - s - F_1(\gamma_1; x, X)^{-1} \bullet \mathrm{Diag}(\mathbf{e} - x)\right)$$
$$+ \frac{1}{\gamma_1}(\mathbf{e} - x)^{\mathsf{T}} \mathrm{diag}(F_1(\gamma_1; x, X)^{-1}(C \circ X)F_1(\gamma_1; x, X)^{-1}) \ .$$

So we have

$$\gamma_1^2 \frac{\partial^2 f_1}{\partial \gamma_1^2} = -n + s + F_1(\gamma_1; x, X)^{-1} \bullet \mathrm{Diag}(\mathbf{e} - x)$$
$$+ \gamma_1(\mathbf{e} - x)^{\mathsf{T}} \mathrm{diag}(F_1(\gamma_1; x, X)^{-1}(C \circ X)F_1(\gamma_1; x, X)^{-1}) \ .$$

Finally, again taking $\psi_1 := \log \gamma_1$, using the chain rule we have

$$\frac{\partial^2 f_1}{\partial \psi_1^2} = \frac{\partial \overline{g}_1}{\partial \psi_1} = \gamma_1 \frac{\partial \overline{g}_1}{\partial \gamma_1} = \gamma_1 \left(\frac{\partial f_1}{\partial \gamma_1} + \gamma_1 \frac{\partial^2 f_1}{\partial \gamma_1^2}\right) = \gamma_1 \frac{\partial f_1}{\partial \gamma_1} + \gamma_1^2 \frac{\partial^2 f_1}{\partial \gamma_1^2}$$
$$= n - s - F_1(\exp(\psi_1); x, X)^{-1} \bullet \mathrm{Diag}(\mathbf{e} - x)$$
$$- n + s + F_1(\exp(\psi_1); x, X)^{-1} \bullet \mathrm{Diag}(\mathbf{e} - x)$$
$$+ \exp(\psi_1)(\mathbf{e} - x)^{\mathsf{T}} \mathrm{diag}(F_1(\exp(\psi_1); x, X)^{-1}(C \circ X)F_1(\exp(\psi_1); x, X)^{-1})$$
$$= \exp(\psi_1)(\mathbf{e} - x)^{\mathsf{T}} \mathrm{diag}(F_1(\exp(\psi_1); x, X)^{-1}(C \circ X)F_1(\exp(\psi_1); x, X)^{-1}) \ .$$

It remains to demonstrate that this last expression is nonnegative. We have $C \succ 0$ and $X \succeq 0$, and therefore $C \circ X \succeq 0$ (Zhang, 2005, page 175). Then, it is also clear from (2.5) that $F_1(\exp(\psi_1; x, X) \succ 0$. Therefore

$$F_1(\exp(\psi_1); x, X)^{-1}(C \circ X)F_1(\exp(\psi_1); x, X)^{-1} \succeq 0.$$

So we have

$$\frac{\partial^2 f_1}{\partial \psi_1^2} \geq 0,$$

and we can conclude that $f_1(\exp(\psi_1); x, X)$ is convex in ψ_1.

Similarly, $f_2(\exp(\psi_2); x, X)$ is convex in ψ_2. Finally, for fixed α and (x, X), $F(\alpha, \exp(\psi_1), \exp(\psi_2); x, X)$ is jointly convex in ψ_1 and ψ_2 because it is a nonnegative weighted sum of $f_1(\exp(\psi_1); x, X)$ and $f_2(\exp(\psi_2); x, X)$ □

Remark. *By working with $\psi = (\psi_1, \psi_2) := (\log(\gamma_1), \log(\gamma_2)) \in \mathbb{R}^2$ and establishing convexity, under smoothness assumptions, we are able to rigorously find the best γ (for a given α) using our BFGS-based Algorithm 2. Moreover, even in an unmixed setting, where we only want to optimize the single scaling parameter for the BQP bound or the complementary BQP*

bound, Algorithm 2 applies, and we get a convergent algorithm (under smoothness assumptions). We note that (Anstreicher, 2018) does not work that way; he applies an approximate Newton's algorithm, working directly with the scaling parameters γ_i (separately for the BQP bound and the complementary BQP bound), approximating the relevant second derivative, and it is not clear that this converges.

2.3.3.2 Optimizing the parameters

The (strengthened) mBQP bound depends on the parameters $(\alpha, \gamma_1, \gamma_2)$. We do not have any type of full joint convexity. But based on Theorem 2.9, to find a good upper bound, we are motivated to formulate two convex problems.

First, for given $\hat{\psi}_1$ and $\hat{\psi}_2$, we consider the convex optimization problem

$$\min\{V_{\hat{\psi}_1,\hat{\psi}_2}(\alpha) \ : \ \alpha \in [0,1]\} \ , \tag{2.7}$$

where

$$\begin{aligned} V_{\hat{\psi}_1,\hat{\psi}_2}(\alpha) &:= \check{v}(C,s;\alpha,\exp(\hat{\psi}_1),\exp(\hat{\psi}_2)) \\ &= (1-\alpha)f_1(C,s;\exp(\hat{\psi}_1);x^*,X^*) + \alpha f_2(C,s;\exp(\hat{\psi}_2);x^*,X^*), \end{aligned}$$

and $(x^*, X^*) = (x^*(\alpha), X^*(\alpha))$ solves the maximization problem in Proposition 2.7 for the given α, when $\gamma_1 = \exp(\hat{\psi}_1)$, and $\gamma_2 = \exp(\hat{\psi}_2)$.

We solve (2.7) with the logarithmic barrier method described in §2.2 and detailed in Algorithm 1.

Then, we also define for given $\hat{\alpha} \in [0,1]$, the convex problem

$$\min\{V_{\hat{\alpha}}(\psi_1,\psi_2) \ : \ (\psi_1,\psi_2) \in \mathbb{R}^2\} \ , \tag{2.8}$$

where

$$\begin{aligned} V_{\hat{\alpha}}(\psi_1,\psi_2) &:= \check{v}(C,s;\hat{\alpha},\exp(\psi_1),\exp(\psi_2)) \\ &= (1-\hat{\alpha})f_1(C,s;\exp(\psi_1);x^*,X^*) + \hat{\alpha} f_2(C,s;\exp(\psi_2);x^*,X^*), \end{aligned}$$

and $(x^*, X^*) = (x^*(\psi_1,\psi_2), X^*(\psi_1,\psi_2))$ solves the maximization problem in Proposition 2.7 for $\alpha = \hat{\alpha}$, $\gamma_1 = \exp(\psi_1)$, and $\gamma_2 = \exp(\psi_2)$.

We solve (2.8) with the quasi-Newton method described in §2.2 and detailed in Algorithm 2. The subgradients g_{ψ_i} , $i = 1, 2$ used in Algorithm 2 are specifically given by

$$g_{\psi_1} = \frac{\partial f_1(\exp(\psi_1); x^*, X^*)}{\partial \psi_1} = n - s - F_1(\exp(\psi_1); x^*, X^*)^{-1} \bullet \mathrm{Diag}(\mathbf{e} - x^*),$$
$$g_{\psi_2} = \frac{\partial f_2(\exp(\psi_2); x^*, X^*)}{\partial \psi_2} = s - F_2(\exp(\psi_2); x^*, X^*)^{-1} \bullet \mathrm{Diag}(x^*).$$

Next, we briefly establish a 'unique-optimum property' for the BQP relaxation that is relevant to the behavior of Algorithms 1 and 2; see Remark 2.2. Let

$$B(\gamma; X) := \gamma C \circ X + I - \mathrm{Diag}(X),$$

for $\gamma > 0$, $(\mathrm{diag}(X), X) \in P(n, s)$.

Lemma 2.10. *Suppose that $C \succ 0$ and $\gamma > 0$. Then*

$$B(\gamma; X) \succ 0, \quad \text{for all} \quad (\mathrm{diag}(X), X) \in P(n, s).$$

Proof. Let $x := \mathrm{diag}(X)$. For $0 \leq x < \mathbf{e}$, we have $\mathrm{Diag}(\mathbf{e} - x) \succ 0$ and $\gamma C \circ X \succeq 0$.

Now suppose that, $x_i = 1$, for $i \in U \subset N$, and $x_i < 1$, for $i \in V := N \setminus U$. After a symmetric permutation of indices we can write

$$\begin{aligned} &\gamma C \circ X + \mathrm{Diag}(\mathbf{e} - x) \\ &\quad = \gamma \begin{pmatrix} C_{UU} \circ X_{UU} & C_{UV} \circ X_{UV} \\ (C_{UV} \circ X_{UV})^{\mathsf{T}} & C_{VV} \circ X_{VV} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathrm{Diag}(\mathbf{e} - x_V) \end{pmatrix}. \end{aligned}$$

We have $\gamma C \circ X + \mathrm{Diag}(\mathbf{e} - x) \succeq 0$ (because $\gamma > 0$, $C \succ 0$, $X \succeq 0$, and $x \leq \mathbf{e}$). By Oppenheim's inequality, we have

$$\det(\gamma C_{UU} \circ X_{UU}) \geq \det(\gamma C_{UU}) \prod_{i \in U} x_i = \det(\gamma C_{UU}) > 0.$$

This together with $\gamma C_{UU} \circ X_{UU} \succeq 0$ implies that

$$\gamma C_{UU} \circ X_{UU} \succ 0. \tag{2.9}$$

Next, we calculate the Schur complement

$$\gamma C_{VV} \circ X_{VV} - (\gamma C_{UV} \circ X_{UV})^{\mathsf{T}} (\gamma C_{UU} \circ X_{UU})^{-1} (\gamma C_{UV} \circ X_{UV}) \succeq 0,$$

by (Zhang, 2005, Theorem 1.12). Hence

$$\gamma C_{VV} \circ X_{VV} - (\gamma C_{UV} \circ X_{UV})^{\mathsf{T}} (\gamma C_{UU} \circ X_{UU})^{-1} (\gamma C_{UV} \circ X_{UV}) + \operatorname{Diag}(\mathbf{e} - x_V) \succ 0. \tag{2.10}$$

Then, from (2.9) and (2.10), considering again (Zhang, 2005, Theorem 1.12), we have

$$\gamma C \circ X + \operatorname{Diag}(\mathbf{e} - x) \succ 0.$$

□

Theorem 2.11. *Suppose that $C \succ 0$ and $\gamma > 0$. If $\gamma C - I$ has no zero entries, then* $\operatorname{ldet} B(\gamma; \cdot)$ *is strictly concave for* $(\operatorname{diag}(X), X) \in P(n, s)$*, and therefore the BQP relaxation has a unique optimal solution.*

Proof. It is easy to verify that the inverse function of $B(\gamma; \cdot)$ is

$$G(\gamma; Y) := (\gamma C - I)^{(-1)} \circ (Y - I),$$

where the superscript '(-1)' denotes Hadamard inverse (i.e., element-wise reciprocals). So clearly the condition for the inverse to be defined is that $\gamma C - I$ has no zero entries. Therefore, $B(\gamma; \cdot)$ is a one-to-one function.

Now, for each $(\operatorname{diag}(X), X) \in P(n, s)$, we can write

$$\operatorname{ldet} B(\gamma; X) = \max_{Z \succ 0} \{\operatorname{ldet} Z \mid Z \preceq B(\gamma; X)\} . \tag{2.11}$$

Considering that $B(\gamma; X) \succ 0$ (Lemma 2.10) and the monotonicity of $\log(\cdot)$, we have that the maximizing $Z \succ 0$ satisfying the equality in the constraint in (2.11) gives the value of $\operatorname{ldet} B(\gamma; X)$, for each $(\operatorname{diag}(X), X) \in P(n, s)$. We note that the constraint $Z \succ 0$ can be relaxed to the positive semidefinite constraint $Z \succeq 0$ without changing the optimal solution and the corresponding value of (2.11). Therefore, as the constraint set in (2.11) with the relaxed constraint $Z \succeq 0$ is compact for any given $B(\gamma; X) \succ 0$, and $\operatorname{ldet} : \mathbb{S}^n_{++} \to \mathbb{R}$ is strictly concave (Bakonyi and Woerdeman, 2011, Corollary 1.4.2), we have that the optimal solution of (2.11), denoted in the following by $Z^*(X)$, exists and is unique for all $(\operatorname{diag}(X), X) \in P(n, s)$.

Then, considering the linearity of $B(\gamma; \cdot)$, the optimality of $Z^*(\cdot)$, and the strict concavity

of $\operatorname{ldet}(\cdot)$, we have

$$\begin{aligned}
&\operatorname{ldet} B(\gamma; \tau X^1 + (1-\tau)X^2) \\
&\quad = \max_{Z \succ 0}\{\operatorname{ldet} Z \mid Z \preceq B(\gamma; \tau X^1 + (1-\tau)X^2)\} \\
&\quad = \max_{Z \succ 0}\{\operatorname{ldet} Z \mid Z \preceq \tau B(\gamma; X^1) + (1-\tau)B(\gamma; X^2)\} \\
&\quad \geq \operatorname{ldet}(\tau Z^*(X^1) + (1-\tau)Z^*(X^2)) \\
&\quad > \tau \operatorname{ldet} Z^*(X^1) + (1-\tau)\operatorname{ldet} Z^*(X^2) \quad \text{(because } B(\gamma;\cdot) \text{ is one-to-one)},
\end{aligned}$$

for all $\tau \in (0,1)$ and $X^1 \neq X^2$. We conclude that $\operatorname{ldet} B(\gamma;\cdot)$ is strictly concave for $(\operatorname{diag}(X), X) \in P(n,s)$. □

Remark. *We note that for any $C \succ 0$ with all entries nonzero, there are at most n values of γ for which the hypothesis of Theorem 2.11 fails to hold; specifically, $\gamma := 1/C_{ii}$, for $i \in N$.*

Finally, in order to obtain a good bound, we propose an algorithmic approach where we start from given values for the parameters α, ψ_1, and ψ_2 and alternate between solving problems (2.7) and (2.8), applying respectively, the procedures described in Algorithms 1 and 2.

In Figure 2.4 we illustrate how f_1, f_2, and the (strengthened) mBQP bound $\check{v}$ vary with each of the parameters ψ_1, and ψ_2, separately, for the instance with $n = 63$, $s = 10$. To construct the first plot in Figure 2.4, we fix α and ψ_2, and vary ψ_1. In the second plot, we fix α and ψ_1 and vary ψ_2. The values of the two parameters that are fixed were obtained by the procedure described above, i.e., alternating between the execution of Algorithms 1 and 2. The interval in which the third parameter varies in each plot is centered at the value also obtained with the alternating algorithm; so the best bound obtained by the algorithm is depicted in the figure. As the mBQP relaxation has a unique optimal solution, $\check{v}$ is differentiable with respect to ψ_1 and ψ_2. In this case, the subgradient presented in Proposition 2.3 is the gradient, and $\partial\check{v}/\partial\psi_i$ is equal to $\alpha_i \partial f_i/\partial\psi_i$, for $i = 1, 2$. These identities are illustrated in Figure 2.4. The gradient of $\check{v}$ is used in Algorithm 2 to optimize the selection of the parameters.

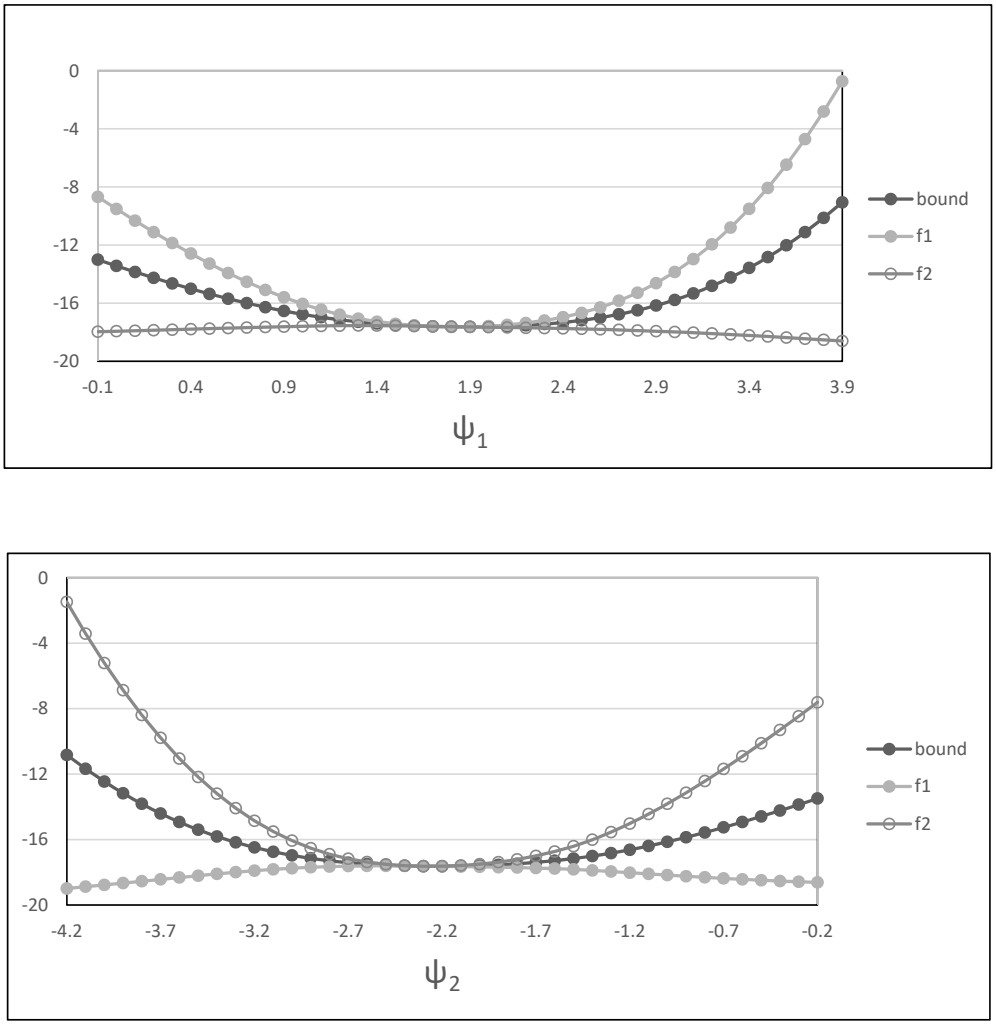

Figure 2.4: Variation of f_1, f_2, and the (strengthened) mBQP, with ψ_1, ψ_2 ($n = 63, s = 10$)

2.4 Mixing the NLP bound with the complementary NLP bound

With the (scaled) NLP bound in mind, we introduce the *mixed NLP (mNLP) bound*:

$$
\begin{aligned}
& w(C, s; \alpha, \gamma_1, \gamma_2) := \\
& \max\ (1-\alpha)\left(\text{ldet}\left(\gamma_1 X^{p/2}(C-D)X^{p/2} + (\gamma_1 D)^x\right) - s\log\gamma_1\right) \\
& \quad + \alpha\left(\text{ldet}\left(\gamma_2 Y^{\bar{p}/2}(C^{-1}-\bar{D})Y^{\bar{p}/2} + (\gamma_2 \bar{D})^y\right) - (n-s)\log\gamma_2 + \text{ldet}\, C\right), \\
& \text{subject to:} \\
& \mathbf{e}^{\mathsf{T}} x = s,\ x + y = \mathbf{e},
\end{aligned}
$$

where $0 \leq \alpha \leq 1$ is a weighting parameter.

The objective function of the mNLP relaxation is defined over the order-n diagonal matrices $D := \text{Diag}(d_1, \ldots, d_n)$ and $\bar{D} := \text{Diag}(\bar{d}_1, \ldots, \bar{d}_n)$, the order-$n$ vectors p and $\bar{p}$, and the scaling parameters $\gamma_1, \gamma_2 > 0$. The following notation is also employed in its definition: $X := \text{Diag}(x)$, $Y := \text{Diag}(y)$, and $(V^u)_{i,i} := V_{i,i}^{u_i}$, $i = 1, \ldots, n$, for a diagonal matrix V and a vector u.

In (Anstreicher, Fampa, Lee, and Williams, 1999), three different strategies are presented for choosing D, p, and γ_1, in order to have the NLP relaxation proven to be a convex program. Analogously, the strategies also apply to the selection of the parameters $\bar{D}$, $\bar{p}$, and γ_2, for the complementary problem. In our numerical experiments with the NLP bound, we have chosen these parameters based on the so-called "NLP-Trace" strategy, where D minimizes the trace of $D-C$, subject to $D-C$ being positive semidefinite. Once D is chosen, the scaling parameter γ_1 should be selected in the interval $[1/d_{\max}, 1/d_{\min}]$ (Anstreicher, Fampa, Lee, and Williams, 1999). In our experiments, we have tested 100 values for γ_1 in this interval and report results for the best one. The same strategy is applied to the complementary problem. Once D and γ_1 ($\bar{D}$ and γ_2) are fixed, the parameter p ($\bar{p}$) can be determined to generate the best possible bound (Anstreicher, Fampa, Lee, and Williams, 1999, Eq. (15)). We note that the optimal scaling factors for the mBQP bound were obtained with quasi-Newton steps in the previous section, as described in Algorithm 2. The same methodology could not be applied here, because the objective function of the mNLP relaxation is neither convex in the scaling parameters nor in the logarithms of the scaling parameters. Therefore, for the results we present on the mNLP bound, we choose γ_1 to be the best scaling parameter for the original NLP bound ($\alpha = 0$), among the 100 values tested, we choose γ_2 to be the best scaling parameter for the complementary NLP bound ($\alpha = 1$), among the 100 values tested. Once γ_1 and γ_2 are chosen, we apply Algorithm 1 to select α for each instance. The results

reported correspond to the optimal α obtained.

Finally, we note that unlike the mBQP bound, the mNLP bound cannot be computed by SDPT3, via Matlab and Yalmip. So, to compute it, we have coded an interior-point algorithm, also in Matlab. The solution procedure is the same as described in (Anstreicher, Fampa, Lee, and Williams, 1999, Section 3), where the NLP bound and the complementary NLP bound are considered. Later, the procedure was also applied in the related work (Anstreicher, Fampa, Lee, and Williams, 2001). The procedure employs a long-step path following methodology, using logarithmic barrier terms for the bound constraints on x (i.e., $0 \leq x \leq e$). For a fixed value of the barrier parameter μ, the barrier function is approximately minimized on $\{x \in \mathbb{R}^n : \mathbf{e}^\mathsf{T} x = s\}$. The parameter μ is then reduced and the process is repeated, until μ is small enough for an approximate minimizer to be within a prescribed tolerance of optimality. The tolerance is certified by a dual solution generated by the algorithm, providing a valid upper bound for the optimal value of NLP.

Next, we briefly establish the 'unique-optimum property' that is relevant to the behavior of Algorithms 1 and 2; see Remark 2.2.

Proposition 2.12. *Let $D \succ C$, $p_i \geq 1$, $0 < d_i \leq \exp(p_i - \sqrt{p_i})$, for $i \in N$. Then*

$$f(x) := \operatorname{ldet}\left(X^{p/2}(C - D)X^{p/2} + (D)^x\right)$$

is strictly concave for $0 < x \leq \mathbf{e}$.

Proof. The result follows directly by replacing $D \succeq C$ by $D \succ C$ in (Anstreicher, Fampa, Lee, and Williams, 1999, Theorem 1). □

Corollary 2.13. *If the hypotheses in Proposition 2.12 are satisfied for the NLP relaxation and for its complement, then the associated mNLP relaxation has a unique optimal solution.*

In Figure 2.5, we illustrate our approach. By mixing the NLP-Trace bound and the complementary NLP-Trace bound, we were able to obtain an improvement for the $n = 124$ problem in the vicinity of $s = 73$.

2.5 On the linx bound and mixing with it

In this section, we take a different notation from scaled linx for simplicity. Formally, we define the *(scaled) linx bound* is

$$\max\left\{\frac{1}{2}v(\gamma; x) \,|\, \mathbf{e}^\mathsf{T} x = s,\, 0 \leq x \leq \mathbf{e}\right\}, \tag{2.12}$$

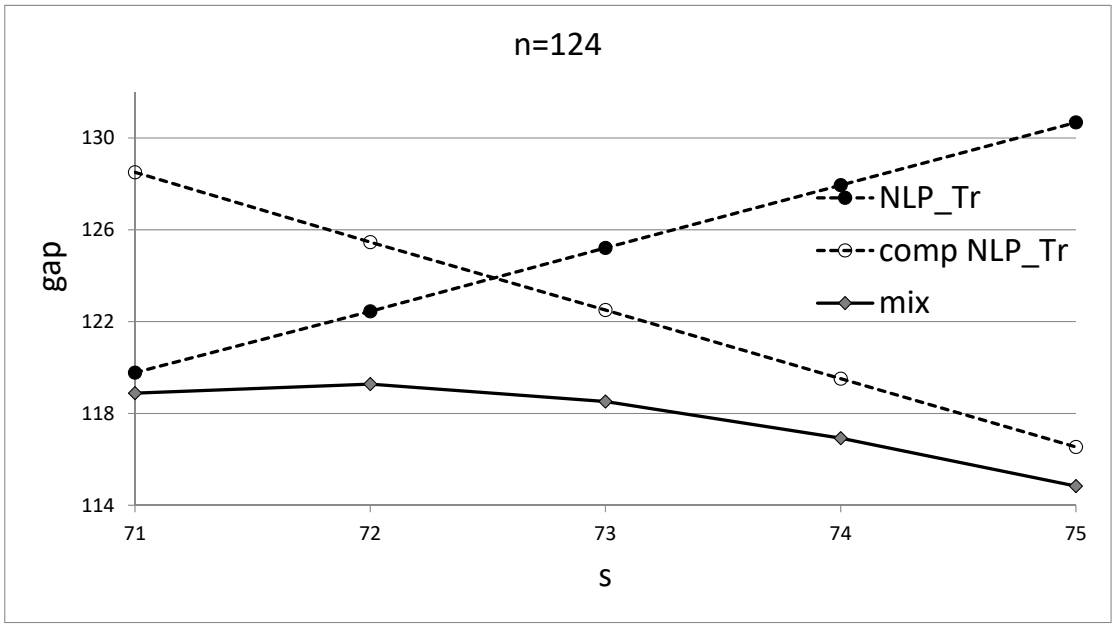

Figure 2.5: Mixing the NLP bound with complementary NLP bound

where

$$v(\gamma; x) := \operatorname{ldet} F(\gamma; x) - s \log \gamma,$$

and

$$F(\gamma; x) := \gamma C \operatorname{Diag}(x) C + \operatorname{Diag}(\mathbf{e} - x).$$

The linx bound has excellent performance, and it is a challenge to improve upon it. In the remainder of this section, we consider fine tuning the bound via its scaling parameter. In §2.5.2, we are able to get an improvement on the linx bound by mixing it both with the NLP bound and with the BQP bound. Note that we do not consider mixing the linx bound with the "complementary linx bound" because the linx bound with scaling parameter γ_1 is equivalent to the complementary linx bound with scaling parameter $\gamma_2 = 1/\gamma_1$.

2.5.1 Optimizing the linx bound on the scaling parameter γ

The linx bound depends on the scaling parameter γ. (Anstreicher, 2020) observed that the linx bound is particularly sensitive to the choice of γ. This is probably due to the fact that the bound is derived by bounding the *square* of the determinant of an order-s principal submatrix of C. So for mixing with the linx bound, it is very useful to be able to efficiently optimize on γ.

To find the best bound, we now define $\psi := \log(\gamma)$ and formulate the problem

$$\min_{\psi} \{H(\psi)\} , \tag{2.13}$$

where

$$H(\psi) := v(\exp(\psi); x^*) ,$$

and where x^* is a maximizer of (2.12), with $\gamma(= \exp(\psi))$ fixed.

Theorem 2.14. *The function $H(\psi)$ is convex in $\psi \in \mathbb{R}$.*

Proof. Based on the same argument used in the proof of Theorem 2.9, we show that $v(\exp(\psi); x)$ is convex in ψ, for fixed x in the feasible set of (2.12).

We have

$$\begin{aligned}
\tfrac{\partial}{\partial\gamma} v(\gamma; x) &= F(\gamma; x)^{-1} \bullet (C \operatorname{Diag}(x) C) - \tfrac{s}{\gamma} \\
&= \tfrac{1}{\gamma}(F(\gamma; x)^{-1} \bullet (F(\gamma; x) + \operatorname{Diag}(x - \mathbf{e})) - s) \\
&= \tfrac{1}{\gamma}(F(\gamma; x)^{-1} \bullet \operatorname{Diag}(x - \mathbf{e}) + n - s) \\
\tfrac{\partial^2}{\partial\gamma^2} v(\gamma; x) &= \tfrac{\partial}{\partial\gamma}\left(\tfrac{1}{\gamma}(F(\gamma; x)^{-1} \bullet \operatorname{Diag}(x - \mathbf{e}) + n - s)\right) \\
&= -\tfrac{1}{\gamma^2}(F(\gamma; x)^{-1} \bullet \operatorname{Diag}(x - \mathbf{e}) + n - s) \\
&+ \tfrac{1}{\gamma}(\mathbf{e} - x)^{\mathsf{T}} \operatorname{diag}(F(\gamma; x)^{-1}(C \operatorname{Diag}(x) C) F(\gamma; x)^{-1}).
\end{aligned}$$

Therefore

$$\begin{aligned}
\tfrac{\partial}{\partial\psi} v(\gamma; x) &= \gamma \tfrac{\partial}{\partial\gamma} v(\gamma; x) \\
&= F(\gamma; x)^{-1} \bullet \operatorname{Diag}(x - \mathbf{e}) + n - s,
\end{aligned}$$

and

$$\begin{aligned}
\tfrac{\partial^2}{\partial\psi^2} v(\gamma; x) &= \gamma \tfrac{\partial}{\partial\gamma} v(\gamma; x) + \gamma^2 \tfrac{\partial^2}{\partial\gamma^2} v(\gamma; x) \\
&= \gamma(\mathbf{e} - x)^{\mathsf{T}} \operatorname{diag}(F(\gamma; x)^{-1}(C \operatorname{Diag}(x) C) F(\gamma; x)^{-1}).
\end{aligned} \tag{2.14}$$

Now, it remains to show that

$$\frac{\partial^2}{\partial\psi^2} v(\exp(\psi); x^*) \geq 0, \ \forall\psi.$$

Considering (2.14), it suffices to show that

$$\operatorname{diag}(F(\exp(\psi); x^*)^{-1}(C \operatorname{Diag}(x^*) C) F(\exp(\psi); x^*)^{-1}) \geq 0.$$

We have $C \succ 0$ and $\operatorname{Diag}(x^*) \succeq 0$, therefore $C \operatorname{Diag}(x^*) C \succeq 0$. Then, it is also clear from

(2.5) that $F(\exp(\psi); x^*) \succ 0$ and, therefore,

$$F(\exp(\psi); x^*)^{-1}(C\,\mathrm{Diag}(x^*)C)F(\exp(\psi); x^*)^{-1} \succeq 0,$$

which completes the proof. □

We solve (2.13) with the quasi-Newton method described in §2.2 and detailed in Algorithm 2, for $m = 1$ and $\alpha_1 = 1$. The subgradient g_{ψ_1} , used in Algorithm 2 is specifically given by

$$g_{\psi_1} = \frac{\partial v(\exp(\psi); x^*)}{\partial \psi} = F(\gamma; x^*)^{-1} \bullet \mathrm{Diag}(x^* - \mathbf{e}) + n - s.$$

Remark. *By working with $\psi := \log(\gamma)$ and establishing convexity, under smoothness assumptions, we are able to rigorously find the best value of γ using our BFGS-based Algorithm 2. We note that (Anstreicher, 2018) does not work that way; he applies an approximate Newton's algorithm, working directly with the scaling parameter γ, approximating the relevant second derivative, and it is not clear that this converges.*

Next, we briefly establish the 'unique-optimum property' that is relevant to the behavior of Algorithms 1 and 2; see Remark 2.2.

Lemma 2.15. *Suppose that $C \succ 0$ and $\gamma > 0$. Then*

$$F(\gamma; x) \succ 0, \quad \text{for all} \quad 0 \leq x \leq \mathbf{e}.$$

Proof. For $0 \leq x < \mathbf{e}$, we have $\mathrm{Diag}(\mathbf{e} - x) \succ 0$ and $\gamma C\,\mathrm{Diag}(x)C \succeq 0$.

Now suppose that, $x_i = 1$, for $i \in U \subset N$, and $x_i < 1$, for $i \in V := N \setminus U$. After a symmetric permutation of indices we can write

$$\begin{aligned}
&\gamma C\,\mathrm{Diag}(x)C + \mathrm{Diag}(\mathbf{e} - x) \\
&\quad = \gamma \begin{pmatrix} C_{UU} & C_{UV} \\ C_{UV}^{\mathsf{T}} & C_{VV} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \mathrm{Diag}(x_V) \end{pmatrix} \begin{pmatrix} C_{UU} & C_{UV} \\ C_{UV}^{\mathsf{T}} & C_{VV} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathrm{Diag}(\mathbf{e} - x_V) \end{pmatrix} \\
&\quad = \gamma \begin{pmatrix} C_{UV} \\ C_{VV} \end{pmatrix} \mathrm{Diag}(x_V) \begin{pmatrix} C_{UV}^{\mathsf{T}} & C_{VV} \end{pmatrix} \\
&\qquad + \gamma \begin{pmatrix} C_{UU} \\ C_{UV}^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} C_{UU} & C_{UV} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathrm{Diag}(\mathbf{e} - x_V) \end{pmatrix}. \qquad (2.15)
\end{aligned}$$

(2.15) simplifies to

$$\begin{pmatrix} \gamma C_{UU}^2 & \gamma C_{UU} C_{UV} \\ \gamma C_{UV}^{\mathsf{T}} C_{UU} & \gamma C_{UV}^{\mathsf{T}} C_{UV} + \operatorname{Diag}(\mathbf{e} - x_V) \end{pmatrix},$$

and then applying the Schur determinant formula (Horn and Johnson, 1985, p.21-22), we obtain that its determinant is

$$\begin{aligned} &\det(\gamma C_{UU}^2) \times \det(\gamma C_{UV}^{\mathsf{T}} C_{UV} + \operatorname{Diag}(\mathbf{e} - x_V) - \gamma C_{UV}^{\mathsf{T}} C_{UU} C_{UU}^{-2} C_{UU} C_{UV}) \\ &\quad = \det(\gamma C_{UU}^2) \times \det(\operatorname{Diag}(\mathbf{e} - x_V)) > 0. \end{aligned}$$

Therefore, we conclude that $F(\gamma; x) \succ 0$, for all $0 \leq x \leq \mathbf{e}$. □

Theorem 2.16. *Suppose that $C \succ 0$ and $\gamma > 0$. If $\lambda_i(\gamma C \circ C) \neq 1$ for all $i \in N$, then $\operatorname{ldet} F(\gamma; \cdot)$ is strictly concave for $0 \leq x \leq \mathbf{e}$, and therefore (2.12) has a unique optimal solution.*

Proof. For $x, y \in \mathbb{R}^n$, we have

$$\begin{aligned} &\gamma C \operatorname{Diag}(x) C + \operatorname{Diag}(\mathbf{e} - x) = \gamma C \operatorname{Diag}(y) C + \operatorname{Diag}(\mathbf{e} - y) \Longleftrightarrow \\ &\gamma C \operatorname{Diag}(x - y) C - \operatorname{Diag}(x - y) = 0. \end{aligned} \tag{2.16}$$

Note that

$$\operatorname{diag}(\gamma C \operatorname{Diag}(x - y) C - \operatorname{Diag}(x - y)) = (\gamma C \circ C - I)(x - y).$$

Therefore, if $\gamma C \circ C - I$ has full rank, (2.16) only holds for $x = y$. Equivalently, if $\lambda_i(\gamma C \circ C) \neq 1$, for all $i \in N$, $F(\gamma; \cdot)$ is a one-to-one function.

Now, for each $x \in [0, 1]^n$, we can write

$$\operatorname{ldet} F(\gamma; x) = \max_{Z \succ 0} \{\operatorname{ldet} Z \mid Z \preceq F(\gamma; x)\} .$$

From Lemma 2.15, we have that $F(\gamma; x) \succ 0$, for all $0 \leq x \leq \mathbf{e}$. Then, considering the linearity of $F(\gamma; \cdot)$, the strict concavity of $\operatorname{ldet}(\cdot)$, and the fact that $F(\gamma; \cdot)$ is a one-to-one function, we can use the same argument used in the proof of Theorem 2.11, to verify that $\operatorname{ldet} F(\gamma; \cdot)$ is strictly concave for $0 \leq x \leq \mathbf{e}$. □

Remark. *We note that for any $C \succ 0$, there are at most n values of γ for which the hypothesis of Theorem 2.16 fails to hold; specifically, $\gamma := 1/\lambda_i(C \circ C)$, for $i \in N$.*

2.5.2 Improvements on the linx bound

By mixing the complementary NLP-Trace bound and the linx bound, we were able to obtain an improvement for the $n = 63$ problem in the vicinity of $s = 25$; see Figure 2.6. By mixing the complementary BQP bound and the linx bound, we were able to obtain an improvement for the $n = 63$ problem in the vicinity of $s = 41$; see Figure 2.7. The results shown in the figures, were obtained with α given by Algorithm 1. To obtain the best α, we fixed both scaling parameters in each mixed relaxation. The scaling parameters were obtained separately, each selected to optimize the corresponding unmixed bound. For the complementary NLP-Trace bound, the selection was done by enumerating 100 different values and selecting the best, as explained in §2.4. For the other bounds, the scaling parameter was optimized with Algorithm 2. When mixing the complementary NLP-Trace bound and the linx bound, we solved the relaxation with an interior-point algorithm that we have implemented in Matlab. When mixing the complementary BQP bound and the linx bound, we solve the relaxation with SDPT3, via Matlab and Yalmip.

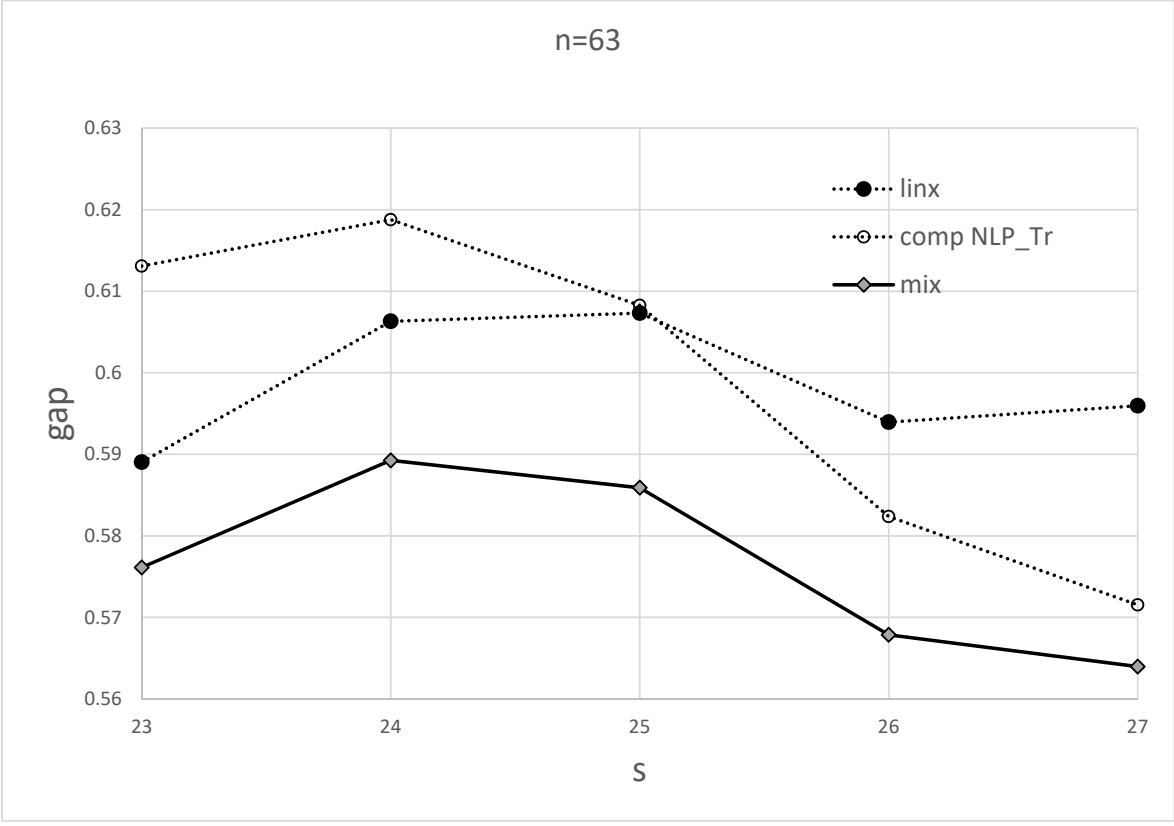

Figure 2.6: Mixing the complementary NLP bound with the linx bound

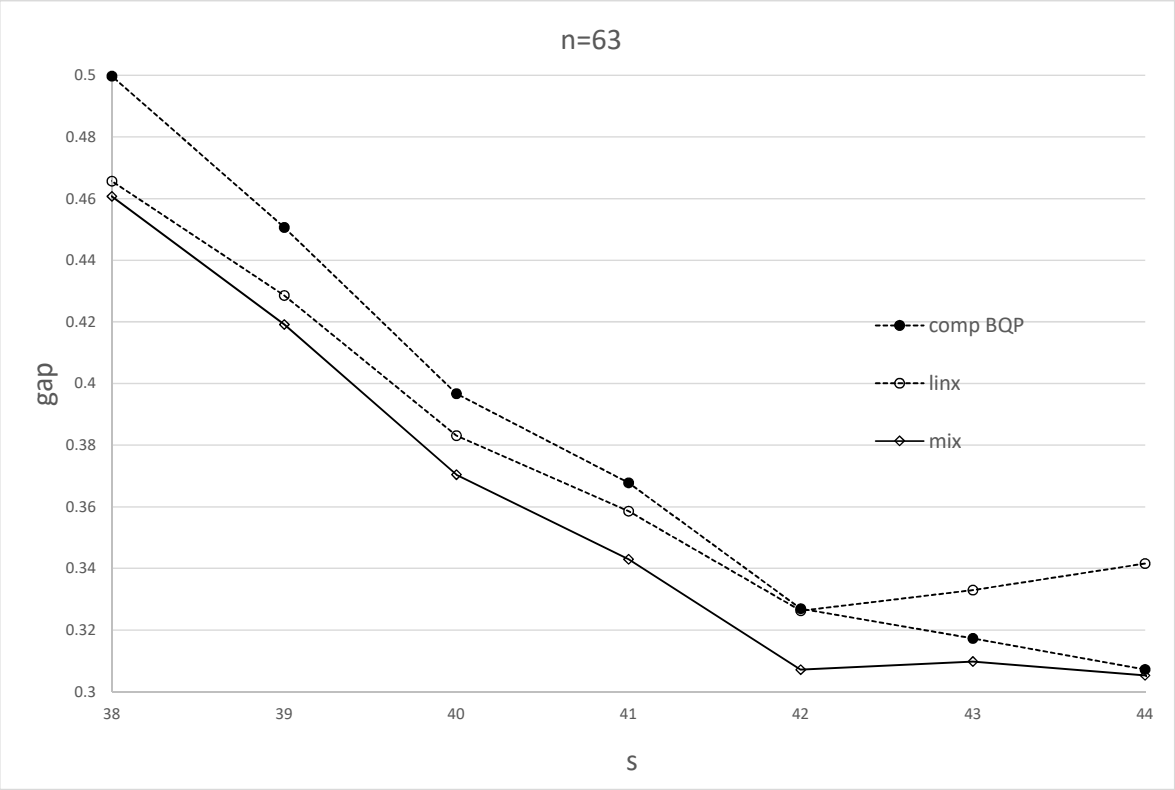

Figure 2.7: Mixing the complementary BQP bound with the linx bound

2.6 Mixing an NLP bound and a BQP bound

A convenient solver for calculating the BQP bound and its complement (and also for calculating the linx bound) is SDPT3 via Yalmip. But the NLP bound and its complement are not amenable to solution by SDPT3 via Yalmip. So we developed our own IPM for calculating the NLP bound. Because of this dichotomy between available solvers, we need a special approach for mixing the NLP bound or its complement, with the BQP bound or its complement.

At this point, we are not very concerned with efficiency. Rather, we only seek a practical method for calculating these particular mixed bounds to see if we can get an improvement on the unmixed bounds by mixing. In this sense, it is a proof of concept for this particular mixing.

Our idea is simply to apply Lagrangian relaxation to the mixing bound, in its form with

duplicated variables, as follows:

$$\begin{aligned} v(\alpha) &:= \max\left\{\alpha f_1(x,\mathcal{X}) + (1-\alpha) f_2(y,\mathcal{Y}) \ : \ (x,\mathcal{X}) \in \mathcal{P},\, (y,\mathcal{Y}) \in \mathcal{Q},\, x+y=e\right\} \\ &= \min_{\pi\in\mathbb{R}^n} \Bigg\{ \max\left\{\alpha f_1(x,\mathcal{X}) + (1-\alpha) f_2(y,\mathcal{Y}) + \pi^{\mathsf{T}}\left(\mathbf{e}-x-y\right) \ : \right. \\ &\qquad\qquad \left. (x,\mathcal{X}) \in \mathcal{P},\ (y,\mathcal{Y}) \in \mathcal{Q}\right\} \Bigg\}. \\ &= \min_{\pi\in\mathbb{R}^n} \Bigg\{ \pi^{\mathsf{T}} e + \max\left\{\alpha f_1(x,\mathcal{X}) - \pi^{\mathsf{T}} x \ : \ (x,\mathcal{X}) \in \mathcal{P}\right\} \\ &\qquad\qquad + \max\left\{(1-\alpha) f_2(y,\mathcal{Y}) - \pi^{\mathsf{T}} y \ : \ (y,\mathcal{Y}) \in \mathcal{Q}\right\} \Bigg\}. \end{aligned}$$

In this form, we apply subgradient optimization to find an optimal $\pi \in \mathbb{R}^n$, and at each step the Lagrangian subproblem decouples into the $(x,\mathcal{X}) \in \mathcal{P}$ maximization problem and the $(y,\mathcal{Y}) \in \mathcal{Q}$ maximization problem. So we can apply separate solvers to each.

In Figure 2.8, we illustrate some success with our approach. For the $n = 124$ problem, we were able to successfully mix the NLP-Trace bound and the complementary BQP bound, in the vicinity of $s = 51, 52$, to obtain an improvement over either bound alone.

2.7 Mixing across a family of instances

So far, we have carried out computations aimed at demonstrating the applicability of our mixing idea across many different bounds for MESP. In this section, we report on an experiment aimed at demonstrating that our ideas can be fruitful across a family of instances. We generated 10 dense random instances of MESP, using the Matlab function `sprandsym()` to obtain our random symmetric matrix C. We chose the eigenvalues of the generated C to be $\lambda_i(C) := M^{\frac{(n+1-i)-i}{n-1}}$, for $i = 1, \ldots, n$, for some $M > 0$. This gives us a nice convex sequence of decreasing eigenvalues, with the added property that C and C^{-1} have the same eigenvalues. In this way, C and C^{-1} are sampled from the same distribution, and we might expect bounding methods to behave similarly on them when s is chosen to be near $n/2$. For each instance, we consistently see modest but significant bound improvements for an s near $n/2$. We note that improvements of this magnitude can lead to a significant increase in the ability to prune subproblems in a branch-and-bound search. For our experiments, we chose $n := 51$ and $M := 4$, and we mixed the NLP bound with the complementary NLP bound (for both, we use the "NLP-Ident" strategy, which comes with a unique scaling parameter;

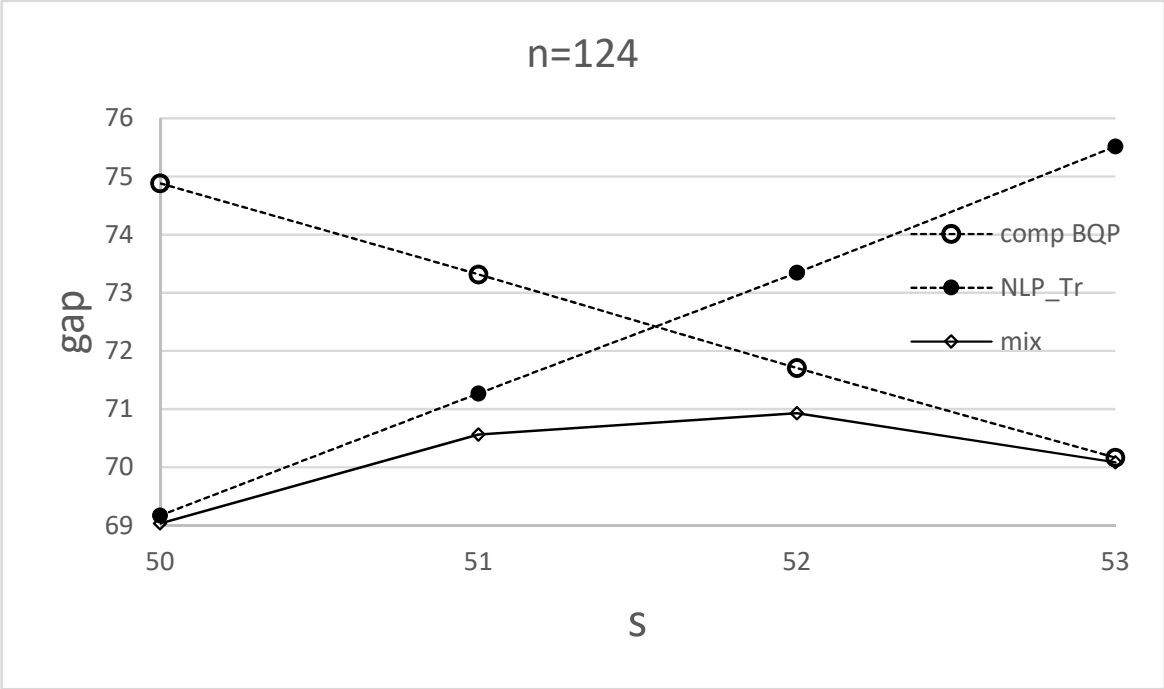

Figure 2.8: Mixing the NLP bound with the complementary BQP bound

see (Anstreicher, Fampa, Lee, and Williams, 1999)), using the optimal mixing parameter α^*. Our results are summarized in Table 2.1, where 'Improvement' (%) is the improvement in gap relative to the best of the two unmixed bounds.

2.8 Concluding remarks

It is a challenge to *efficiently* employ our ideas in the context of branch-and-bound. We need to find effective mixing parameters α quickly. Note that in our notation, (Anstreicher, 2018) is using $m = 2$ and $\alpha = 0$ or 1, and in the context of branch-and-bound, each child inherits α from its parent, only updating the choice occasionally. In the context of branch-and-bound, we would now expect that for many subproblems, we would have $\alpha = 0$ or 1. But we can further expect that for many we would have $0 < \alpha < 1$, and we would then gain from our approach. The guidance of (Anstreicher, 2018) is: "we use a simple criterion based on the number of fixed variable and depth in the tree to decide when to check the other bound". So we would proceed similarly, doing a search for a good α (see §2.2) after an inherited value becomes stale.

It is not clear at all how our mixing idea could be adapted to "spectral and masked

Inst	s	gap ($\alpha = 0$)	gap ($\alpha = 1$)	gap (α^*)	α^*	Improvement (%)
1	24	4.1160	5.0785	4.1160	0	0
1	25	4.3390	4.6437	4.2886	0.25	1.1610
1	26	4.5991	4.2445	4.2044	0.78	0.9458
2	24	4.1762	5.1444	4.1762	0	0
2	25	4.4746	4.7894	4.4074	0.28	1.5031
2	26	4.9186	4.5778	4.5007	0.77	1.6847
3	24	3.9669	5.1077	3.9669	0	0
3	25	4.2585	4.7247	4.2526	0.08	0.1387
3	26	4.6242	4.4146	4.3635	0.71	1.1575
4	24	4.1022	5.3773	4.1022	0	0
4	25	4.1731	4.7654	4.1303	0.17	1.0249
4	26	4.4525	4.3601	4.1708	0.55	4.3419
5	24	3.8534	5.1652	3.8534	0	0
5	25	4.1871	4.8372	4.1871	0	0
5	26	4.6224	4.6080	4.5131	0.55	2.0605
6	24	3.9116	4.8674	3.9116	0	0
6	25	4.2430	4.5111	4.1577	0.31	2.0101
6	26	4.5895	4.1688	4.1262	0.78	1.0217
7	24	3.5030	4.0796	3.4829	0.14	0.5754
7	25	3.8599	3.7607	3.6162	0.57	3.8426
7	26	4.2283	3.4520	3.4508	0.96	0.0357
8	24	4.4715	4.9430	4.4281	0.18	0.9716
8	25	4.8435	4.6521	4.5153	0.70	2.9389
8	26	5.2018	4.3473	4.3382	0.92	0.2095
9	24	4.5169	5.5164	4.5169	0	0
9	25	4.7287	5.0820	4.7055	0.21	0.4910
9	26	4.9817	4.6871	4.6444	0.75	0.9107
10	24	3.2562	4.0135	3.2515	0.06	0.1432
10	25	3.7414	3.8177	3.5976	0.44	3.8449
10	26	4.1017	3.4959	3.4757	0.89	0.5759

Table 2.1: Mixing the NLP with the complementary NLP bound ($n = 51$)

spectral bounds" (Ko, Lee, and Queyranne, 1995; Anstreicher and Lee, 2004; Burer and Lee, 2007; Hoffman, Lee, and Williams, 2001; Lee and Williams, 2003), because these are apparently not based on convex relaxation. We would like to highlight this as an interesting area to explore.

Very recently, (Li and Xie, 2023) presented new results on a relaxation and on an approximation algorithm for MESP. It would be interesting to see if some of those results can be exploited in our context.

Finally, our general mixing idea, although well suited for MESP, should find application to other combinatorial-optimization problems with nonlinearities. It is a challenge to find other good applications.

CHAPTER 3

On Computing with some Convex Relaxations for the Maximum-Entropy Sampling Problem

This chapter has been published as:

Zhongzhu Chen, Marcia Fampa, Jon Lee. On computing with some convex relaxations for the maximum-entropy sampling problem. *INFORMS Journal on Computing*, 35(2):368-385, 2023. https://doi.org/10.1287/ijoc.2022.1264

3.1 Introduction

In this chapter, based on a factorization of an input covariance matrix, we define a mild generalization of an upper bound of (Nikolov, 2015) and (Li and Xie, 2023) for CMESP. We demonstrate that this factorization bound is invariant under scaling and also independent of the particular factorization chosen. We give a variable-fixing methodology that could be used in a branch-and-bound scheme based on the factorization bound for exact solution of CMESP, and we demonstrate that its ability to fix is independent of the factorization chosen. We report on successful experiments with a commercial nonlinear-programming solver. We further demonstrate that the known "mixing" technique (chapter 2) can be successfully used to combine the factorization bound with the factorization bound of the complementary CMESP, and also with the (scaled) linx bound.

In §3.2, we discuss some upper bounds for CMESP. In particular, we present the "factorization bound" for CMESP, as a convex formulation DFact of the Lagrangian dual of a nonconvex formulation Fact, and some of its important properties from a computational point of view. In particular: (i) We demonstrate that the factorization bound changes by the same amount as the objective of CMESP, when C is scaled by some $\gamma > 0$. (ii) We give a variable fixing methodology based on a feasible solution of DFact. Variable fixing has also been studied for CMESP in the context of other convex relaxations; see (Anstreicher,

Fampa, Lee, and Williams, 1996, 1999, 2001; Anstreicher, 2018, 2020). (iii) We demonstrate that the factorization bound and its ability to fix variables, based on a matrix factorization $C = FF^{\mathsf{T}}$, is independent of the factorization (and so mathematically, it provides the same bound as that of (Nikolov, 2015) and (Li and Xie, 2023)). (iv) We demonstrate that the factorization bound dominates the well-known "spectral bound". (v) Although it is possible to directly attack DFact to calculate the factorization bound, it is much more fruitful to work with DDFact, a convex formulation of the Lagrangian dual of DFact. In connection with this, we provide a mechanism to get a minimum-gap feasible solution of DFact, relative to a feasible (and possibly non-optimal) solution of DDFact (which is useful for getting a true upper bound for CMESP). We also describe how to get the gradient of the objective function of DDFact (under a technical condition), which is necessary for applying any reasonable technique for efficiently solving DDFact. (vi) We review the linx bound for CMESP, and we present some of its key properties that are useful for computing. (vii) We review the "mixing" bound of (Chen, Fampa, Lambert, and Lee, 2021), and we work out a dual for it, as well as a fixing methodology in an important case that generalizes what we can do for the DDFact, complementary DDFact, and linx bounds.

In §3.3, we discuss the numerical experiments where, using a commercial nonlinear-programming solver, we calculate upper bounds for benchmark instances of MESP from the literature with the three relaxations presented, namely DDFact, complementary DDFact and linx, and with the "mixing" strategy described in (Chen, Fampa, Lambert, and Lee, 2021). Generally, we found that a commercial nonlinear-programming solver is quite viable for our relaxations, even for DDFact which may have nondifferentiabilty. Our main findings: (i) We compared integrality gaps given by the difference between the upper bounds computed with the relaxations and the lower bounds computed with a greedy/interchange heuristic or with the optimal value when we could obtain it by branch-and-bound. We found that all the three relaxations, DDFact, complementary DDFact, and linx, achieve the best bounds for some of the instances, however DDFact and linx achieve together most of the best bounds. (ii) We compared the times to solve the relaxations and analyzed the impact of two factors in the times: the smoothness of the objective functions of the relaxations and the ranks of the covariance matrices. The possibility of DDFact and complementary DDFact encountering points at which the objective function is nondifferentiable led us to the application of a BFGS-based algorithm to solve these relaxations, and the smoothness of linx results in the application of a Newton-based algorithm to solve it. We see a better convergence of the Newton-based algorithm, resulting in best times for linx and with less variability among the different values of s, except for our largest instance with $n = 2000$ and a covariance matrix with rank $r = 949$. In this case, linx presents the drawback of dealing

with an order-n matrix in its objective function, while DDFact deals with an order-r matrix. (iii) We demonstrated how the "mixing" procedure can decrease the bounds obtained with the three relaxations when we mix two relaxations that obtain very similar bounds when applied separately. This is mostly observed when we mix DDFact and linx. For the majority of the instances, DDFact presents better bounds for values of s up to an intermediate value, and for larger values of s, linx presents better bounds. For values of s close to this intermediate value, the mixing strategy effectively decreases the bound obtained by each relaxation. (iv) We demonstrated how the fixing methodology can fix a significant number of variables, especially when applied iteratively, to our largest instance.

In §3.4, we summarize our results and point to future work.

3.2 Upper bounds

There are a wide variety of upper bounding methods for CMESP. The tightest bounds, from a practical computational viewpoint, seem to be Anstreicher's "linx bound" (Anstreicher, 2020) and the "factorization bound". In this section, we describe and develop these bounds, with an eye on practical and efficient computation.

3.2.1 Fact

We begin by introducing a mild generalization of a nonconvex programming bound developed by (Nikolov, 2015) and (Li and Xie, 2023). Suppose that the rank of C is $r \geq s$. Then we factorize $C = FF^{\mathsf{T}}$, with $F \in \mathbb{R}^{n\times k}$, for some k satisfying $r \leq k \leq n$. We note that this could be a Cholesky-type factorization (i.e., $k := r$ and F lower triangular), as in (Nikolov, 2015) and (Li and Xie, 2023). But it could alternatively be derived from a spectral decomposition of C; that is, $C = \sum_{\ell=1}^{r} \lambda_\ell v_\ell v_\ell^{\mathsf{T}}$, where we put $\sqrt{\lambda_\ell} v_\ell$ as column ℓ of F, $\ell = 1, \ldots, k := r$. Another very useful possibility is to let $k := n$, and choose F to be the matrix square root, $C^{1/2}$, which is always symmetric.

Next, for $x \in [0,1]^n$, we define $F(x) := \sum_{j\in N} F_{j\cdot}^{\mathsf{T}} F_{j\cdot}\, x_j \; = F^{\mathsf{T}} \operatorname{Diag}(x) F$ and

$$\begin{array}{rll} z_{\text{Fact}}(C, s, A, b; F) := & \max \sum_{\ell=1}^{s} \log\left(\lambda_\ell(F(x))\right) & \\ & \text{subject to:} & \text{(Fact)} \\ & \quad \mathbf{e}^{\mathsf{T}} x = s,\ Ax \leq b,\ 0 \leq x \leq \mathbf{e}. & \end{array}$$

It is easy to check that the objective of CMESP $z(C, s, A, b) \leq z_{\text{Fact}}(C, s, A, b; F)$, for any factorization of C (c.f. (Ko, Lee, and Queyranne, 1995)). Unfortunately, Fact is not a convex program, so it is not practical to work with.

3.2.2 DFact

We define

$$f_{\text{DFact}}(\Theta, \nu, \pi, \tau) := -\sum_{\ell=k-s+1}^{k} \log\left(\lambda_\ell\left(\Theta\right)\right) + \nu^{\mathsf{T}}\mathbf{e} + \pi^{\mathsf{T}} b + \tau s - s,$$

and the *factorization bound*

$$\begin{array}{rll} z_{\text{DFact}}(C, s, A, b; F) := & \min & f_{\text{DFact}}(\Theta, \nu, \pi, \tau) \\ & \text{subject to:} & \\ & & \operatorname{diag}(F\Theta F^{\mathsf{T}}) + \upsilon - \nu - A^{\mathsf{T}}\pi - \tau\mathbf{e} = 0, \\ & & \Theta \succ 0,\ \upsilon \geq 0,\ \nu \geq 0,\ \pi \geq 0. \end{array} \tag{DFact}$$

DFact is equivalent to the Lagrangian dual of Fact, and it is a convex program. The objective function of DFact is analytic at every point $(\hat{\Theta}, \hat{\upsilon}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ for which $\lambda_{k-s}(\hat{\Theta}) > \lambda_{k-s+1}(\hat{\Theta})$. In fact, we have seen in our experiments, a good solver can get to an optimum, even when this condition fails.

It turns out that the factorization bound for MESP has a close relationship with the spectral bound of (Ko, Lee, and Queyranne, 1995): $\sum_{\ell=1}^{s} \log \lambda_\ell(C)$. First, we establish that like the spectral bound for MESP, the factorization bound for CMESP is invariant under multiplication of C by a scale factor γ, up to the additive constant $-s \log \gamma$, a property that is *not* shared with other convex-optimization bounds.

Theorem 3.1. *For all $\gamma > 0$ and factorizations $C = FF^{\mathsf{T}}$, we have*

$$z_{DFact}(C, s, A, b; F) = z_{DFact}(\gamma C, s, A, b; \sqrt{\gamma}F) - s \log \gamma.$$

Proof. We simply observe that for every feasible solution $(\hat{\Theta}, \hat{\upsilon}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ of DFact, we have that $(\frac{1}{\gamma}\hat{\Theta}, \hat{\upsilon}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ is a feasible solution of DFact with F replaced by $\sqrt{\gamma}F$. Then we observe that $\lambda_\ell\left(\frac{1}{\gamma}\hat{\Theta}\right) = \frac{1}{\gamma}\lambda_\ell(\hat{\Theta})$, for all ℓ. This mapping between feasible solutions is a bijection, so the result follows. □

Because the factorization bound shifts by the same amount as $z(C, s, A, b)$, under scaling of C, we cannot improve on the factorization bound by scaling. In contrast, the linx bound is very sensitive to the choice of the scale factor, and while we can compute an optimal scale factor for the linx bound (see (Chen, Fampa, Lambert, and Lee, 2021)), it is a significant computational burden to do so.

Next, we present another useful result that guides practical usage.

Theorem 3.2. *Let $C = F_j F_j^\mathsf{T}$, for $j = 1, 2$, be two different factorizations of C, and let $(\hat{\Theta}_1, \hat{v}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ be a feasible solution to DFact, for $F := F_1$. Then, there is a feasible solution $(\hat{\Theta}_2, \hat{v}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ to DFact, for $F := F_2$, such that $f_{DFact}(\hat{\Theta}_1, \hat{\nu}, \hat{\pi}, \hat{\tau}) = f_{DFact}(\hat{\Theta}_2, \hat{\nu}, \hat{\pi}, \hat{\tau})$.*

Proof. Let r be the rank of C, and let $C = \sum_{\ell=1}^{n} \lambda_\ell u_\ell u_\ell^\mathsf{T}$ be a spectral decomposition of C. Suppose that $C = FF^\mathsf{T}$, with $F \in \mathbb{R}^{n \times k}$, and $r \leq k \leq n$. Our preliminary goal is to build a special singular-value decomposition of F.

Let $\sigma_\ell := \sqrt{\lambda_\ell}$, for $1 \leq \ell \leq k$. Now define $v_\ell \in \mathbb{R}^k$, for $1 \leq \ell \leq r$ by $v_\ell := \frac{1}{\sigma_\ell} F^\mathsf{T} u_\ell$.

We can easily check that for $1 \leq i \leq \ell \leq r$, we have

$$v_i^\mathsf{T} v_\ell = \frac{1}{\sigma_i \sigma_\ell} u_i^\mathsf{T} F F^\mathsf{T} u_\ell = \frac{1}{\sigma_i \sigma_\ell} u_i C u_\ell = \frac{\lambda_\ell}{\sigma_i \sigma_\ell} u_i^\mathsf{T} u_\ell = \begin{cases} 1, & \text{for } i = \ell; \\ 0, & \text{for } i < \ell. \end{cases}$$

That is, $\{v_\ell \ : \ 1 \leq \ell \leq r\}$ is a set of r orthonormal vectors in $\mathbb{R}^k$. So, for $r < \ell \leq k$, we can now choose v_i so as to complete $\{v_\ell \ : \ 1 \leq \ell \leq r\}$ to an orthonormal basis of $\mathbb{R}^k$.

Next, we have $\sum_{\ell=1}^{k} \sigma_\ell u_\ell v_\ell^\mathsf{T} = \sum_{\ell=1}^{r} \sigma_\ell u_\ell v_\ell^\mathsf{T} = \sum_{\ell=1}^{r} u_\ell u_\ell^\mathsf{T} F = \sum_{\ell=1}^{n} u_\ell u_\ell^\mathsf{T} F = I_n F = F$, and so we can conclude that $F = \sum_{\ell=1}^{k} \sigma_\ell u_\ell v_\ell^\mathsf{T}$ is a singular-value decomposition for F.

The important takeaway is that the $u_\ell \in \mathbb{R}^n$ $(1 \leq \ell \leq n)$ and the nonzero σ_ℓ $(1 \leq \ell \leq r)$ in the singular-value decomposition that we constructed for F only depend on C, not on the particular factorization $C = FF^\mathsf{T}$.

It is convenient now to establish that in a factorization matrix F, we can without loss of generality take $k = n$, by appending 0 columns to F if needed, and this will not affect the bound $z_{\text{DFact}}(C, s, A, b; F)$. Let $\bar{F} := [F \mid 0_{n \times (n-k)}]$, and consider

$$\bar{\Theta} = \begin{pmatrix} \hat{\Theta} & \times \\ \times & \times \end{pmatrix} \in \mathbb{S}_+^n \ .$$

It is easy to check that $\bar{F} \bar{\Theta} \bar{F}^\mathsf{T} = F \hat{\Theta} F^\mathsf{T}$. By Cauchy's eigenvalue interlacing inequalities (see (Horn and Johnson, 1985), for example), we have $\lambda_{\ell+n-k}(\bar{\Theta}) \leq \lambda_\ell(\hat{\Theta})$, for $1 \leq \ell \leq k$. Therefore, we have $z_{\text{DFact}}(C, s, A, b; \bar{F}) \geq z_{\text{DFact}}(C, s, A, b; F)$. Conversely, suppose that $\hat{\Theta} \in \mathbb{S}_+^k$. Now define

$$\bar{\Theta} := \begin{pmatrix} \hat{\Theta} & 0^\mathsf{T} \\ 0 & \lambda_1(\hat{\Theta}) I_{n-k} \end{pmatrix} \in \mathbb{S}_+^n \ .$$

As above, we have $\bar{F} \bar{\Theta} \bar{F}^\mathsf{T} = F \hat{\Theta} F^\mathsf{T}$. And by construction, we have $\lambda_{\ell+n-k}(\bar{\Theta}) = \lambda_\ell(\hat{\Theta})$, for $1 \leq \ell \leq k$. And therefore, we have $z_{\text{DFact}}(C, s, A, b; \bar{F}) \leq z_{\text{DFact}}(C, s, A, b; F)$.

With this we can now conclude that if we have two different factorizations of C, say $C = F_j F_j^\mathsf{T}$ for $j = 1, 2$, we can without loss of generality assume for each that F_j has $k = n$ columns, and further that we can choose singular-value decompositions of the form

$F_j = U\Sigma V_j^\mathsf{T}$, where here we now take U, Σ, V_1 and V_2 to all be $n \times n$. Right multiplying $U\Sigma V_2^\mathsf{T} = F_2$ by $V_2 V_1^\mathsf{T}$, we get $U\Sigma V_2^\mathsf{T} V_2 V_1^\mathsf{T} = F_2 V_2 V_1^\mathsf{T}$ and so $F_1 = F_2 V_2 V_1^\mathsf{T}$.

Finally, for $\Theta_1 \succ 0$, we have $F_1\Theta_1 F_1^\mathsf{T} = F_2 V_2 V_1^\mathsf{T}\Theta_1 V_1 V_2^\mathsf{T} F_2^\mathsf{T}$, and so by taking $\Theta_2 := V_2 V_1^\mathsf{T}\Theta_1 V_1 V_2^\mathsf{T}$, we get $F_1\Theta_1 F_1^\mathsf{T} = F_2\Theta_2 F_2^\mathsf{T}$, with Θ_2 being similar to Θ_1. Therefore, we have that $\lambda_\ell(\Theta_1) = \lambda_\ell(\Theta_2)$, for all ℓ, and so we can transform any feasible solution of DFact with respect to factor $F := F_1$ into a feasible solution of DFact having the same objective value with respect to factor $F := F_2$. The result follows. □

Corollary 3.3. *The value of the factorization bound is independent of the factorization.*

Remark. *Cor. 3.3 follows directly from Thm. 3.2. We note that the proof of Thm. 3.2 not only confirms the statement in Cor. 3.3, but also presents a methodology for constructing a feasible solution $(\hat{\Theta}_2, \hat{v}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ to DFact for a given factorization of C, from a feasible solution $(\hat{\Theta}_1, \hat{v}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ to DFact for any other factorization, where both solutions have the same objective value with respect to the corresponding factor. In §3.2.3, we also present a short proof for Cor. 3.3.*

Next, we establish that the factorization bound for MESP dominates the spectral bound for MESP. While the spectral bound is much cheaper to compute, because of this result, there is never any point of computing the spectral bound if we have already computed the factorization bound. In another way of thinking, if the spectral bound comes close to allowing us to discard a subproblem in the context of branch-and bound, it should be well worth computing the factorization bound to attempt to discard the subproblem.

Theorem 3.4. *Let $C \in \mathbb{S}_+^n$, with $r := \mathit{rank}(C)$, and $s \leq r$. Then, for all factorizations $C = FF^\mathsf{T}$, we have $z_{DFact}(C, s, \cdot, \cdot\,; F) \leq \sum_{\ell=1}^{s} \log \lambda_\ell(C)$.*

Proof. Let $C = \sum_{\ell=1}^{n} \lambda_\ell(C) u_\ell u_\ell^\mathsf{T}$ be a spectral decomposition of C. Because $\lambda_\ell = 0$ for $\ell > r$, $C = \sum_{\ell=1}^{n} \lambda_\ell(C) u_\ell u_\ell^\mathsf{T} = \sum_{\ell=1}^{r} \lambda_\ell(C) u_\ell u_\ell^\mathsf{T}$. By Thm. 3.2, it suffices to take F to be the symmetric matrix $\sum_{\ell=1}^{r} \sqrt{\lambda_\ell(C)} u_\ell u_\ell^\mathsf{T}$.

We consider the solution for DFact given by: $\hat{\Theta} := C^\dagger + \frac{1}{\lambda_r(C)}\left(I - CC^\dagger\right)$, where $C^\dagger := \sum_{\ell=1}^{r} \frac{1}{\lambda_\ell(C)} u_\ell u_\ell^\mathsf{T}$ is the Moore-Penrose pseudoinverse of C, $\hat{v} := \mathbf{e} - \mathrm{diag}(F\hat{\Theta}F^\mathsf{T})$, $\hat{\nu} := 0$, $\hat{\pi} := 0$, and $\hat{\tau} := 1$. We can verify that the r least eigenvalues of $\hat{\Theta}$ are $\frac{1}{\lambda_1(C)}, \frac{1}{\lambda_2(C)}, \ldots, \frac{1}{\lambda_r(C)}$ and the $n - r$ greatest eigenvalues are all equal to $\frac{1}{\lambda_r(C)}$. Therefore, $\hat{\Theta}$ is positive definite.

The equality constraint of DFact is clearly satisfied at this solution. Additionally, we can verify that $F\hat{\Theta}F^\mathsf{T} = \sum_{\ell=1}^{r} u_\ell u_\ell^\mathsf{T}$. As the positive semidefinite matrix $\sum_{\ell=r+1}^{n} u_\ell u_\ell^\mathsf{T}$ is equal to $I - \sum_{\ell=1}^{r} u_\ell u_\ell^\mathsf{T}$, we conclude that $\mathrm{diag}(F\hat{\Theta}F^\mathsf{T}) \leq \mathbf{e}$. Therefore, $\hat{v} \geq 0$, and the solution constructed is a feasible solution to DFact. Finally, we can see that the objective value of this solution is equal to the spectral bound. The result then follows. □

Remark. *We note that when C is nonsingular, then $F = C^{1/2}$, and using the symmetry of $C^{1/2}$, it is easy to directly check that with $\hat{\Theta} := C^{-1}$, $\hat{\tau} := 1$, $\hat{\upsilon} := \hat{\nu} := 0$, and $\hat{\pi} := 0$, we have a feasible solution of DFact with objective value equal to the spectral bound.*

Next, we consider variable fixing, in the context of solving CMESP.

Theorem 3.5. *Let*

- *LB be the objective-function value of a feasible solution for CMESP,*
- *$(\hat{\Theta}, \hat{\upsilon}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ be a feasible solution for DFact with objective-function value $\hat{\zeta}$.*

Then, for every optimal solution x^* for CMESP, we have:

$$x_j^* = 0, \ \forall\ j \in N \text{ such that } \hat{\zeta} - LB < \hat{\upsilon}_j \ ,$$
$$x_j^* = 1, \ \forall\ j \in N \text{ such that } \hat{\zeta} - LB < \hat{\nu}_j \ .$$

Proof. Consider Fact with the additional constraint $x_i = 1$. The dual becomes then,

$$\begin{array}{ll} \min & -\sum_{\ell=k-s+1}^{k} \log\left(\lambda_\ell\left(\Theta\right)\right) + \nu^{\mathsf{T}}\mathbf{e} + \pi^{\mathsf{T}} b + \tau s - s - \omega \\ & \text{subject to:} \\ & \operatorname{diag}(F\Theta F^{\mathsf{T}}) + \upsilon - \nu - A^{\mathsf{T}}\pi - \tau\mathbf{e} + \omega\mathbf{e}_j = 0, \\ & \Theta \succ 0, \ \upsilon \geq 0, \ \nu \geq 0, \ \pi \geq 0. \end{array} \tag{3.1}$$

where ω is the new dual variable. Notice that, as long as $\hat{\upsilon}_j - \omega \geq 0$, $(\hat{\Theta}, \hat{\upsilon} - \omega\mathbf{e}_j, \hat{\nu}, \hat{\pi}, \hat{\tau}, \omega)$ is a feasible solution of the modified dual, with objective value $\hat{\zeta} - \omega$. So, to minimize the objective value of our feasible solution of the modified dual, we set ω equal to $\hat{\upsilon}_j$. We conclude that $\hat{\zeta} - \hat{\upsilon}_j$ is an upper bound on the objective value of every solution of CMESP that satisfies $x_j = 1$. So if $\hat{\zeta} - \hat{\upsilon}_j < LB$, then no optimal solution of CMESP can have $x_j = 1$.

Similarly, consider Fact with the additional constraint $x_j = 0$. In this case, the new dual problem is equivalent to (3.1), except that the objective function does not have the term $-\omega$. Therefore, as long as $\hat{\nu}_j + \omega \geq 0$, $(\hat{\Theta}, \hat{\upsilon}, \hat{\nu} + \omega\mathbf{e}_j, \hat{\pi}, \hat{\tau}, \omega)$ is a feasible solution of this modified dual with objective value $\hat{\zeta} + \omega$, and to minimize the objective value of the feasible solution, we set ω equal to $-\hat{\nu}_j$. Now, we conclude that $\hat{\zeta} - \hat{\nu}_j$ is an upper bound on the objective value of every solution of CMESP that satisfies $x_j = 0$. So if $\hat{\zeta} - \hat{\nu}_j < LB$, then no optimal solution of CMESP can have $x_j = 0$. □

Remark. *Thm. 3.2 implies that all factorizations have the same power to fix variables.*

3.2.3 DDFact

While it turns out that the bound given by DFact is generally quite good, and it has the potential to fix variables at 0/1 values via Thm. 3.5, the model DFact is not easy to solve directly. We instead present its (equivalent) Lagrangian dual, DDFact, which is much easier to work with computationally.

Lemma 3.6. *(see (Nikolov, 2015, Lem. 13)) Let $\lambda \in \mathbb{R}^k_+$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ and let $0 < s \leq k$. With the convention $\lambda_0 = +\infty$, there exists a unique integer ι, with $0 \leq \iota < s$, such that*

$$\lambda_\iota > \tfrac{1}{s-\iota} \sum_{\ell=\iota+1}^{k} \lambda_\ell \geq \lambda_{\iota+1}.$$

Suppose that $\lambda \in \mathbb{R}^k_+$, and assume that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$. Given an integer s with $0 < s \leq k$, let ι be the unique integer defined by Lem. 3.6. We define

$$\phi_s(\lambda) := \sum_{\ell=1}^{\iota} \log(\lambda_\ell) + (s-\iota) \log\left(\tfrac{1}{s-\iota} \sum_{\ell=\iota+1}^{k} \lambda_\ell\right).$$

Next, for $X \in \mathbb{S}^k_+$, we define $\Gamma_s(X) := \phi_s(\lambda_1(X), \ldots, \lambda_k(X))$. Finally, we define

$$\begin{array}{rll} z_{\text{DDFact}}(C, s, A, b; F) := & \max & \Gamma_s(F(x)) \\ & \text{subject to:} & \\ & & \mathbf{e}^\mathsf{T} x = s,\ Ax \leq b,\ 0 \leq x \leq \mathbf{e}. \end{array} \tag{DDFact}$$

It is a result of (Nikolov, 2015) that DDFact is a convex program, and that it is in fact equivalent to the Lagrangian dual of DFact. Checking a Slater's condition, we have that $z_{\text{DDFact}}(C, s, A, b; F) = z_{\text{DFact}}(C, s, A, b; F)$. The advantage of solving DDFact instead of DFact is that it has many fewer variables. But, variable fixing (see Thm. 3.5) relies on a good feasible solution of DFact. Moreover, certifying the quality of a feasible solution of DDFact also requires a good feasible solution of DFact. Motivated by these points, we show how to construct a feasible solution of DFact from a feasible solution $\hat{x}$ of DDFact with finite objective value, with the goal of producing a small gap.

We consider the spectral decomposition $F(\hat{x}) = \sum_{\ell=1}^{k} \hat{\lambda}_\ell \hat{u}_\ell \hat{u}_\ell^\mathsf{T}$, with $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_{\hat{r}} > \hat{\lambda}_{\hat{r}+1} = \cdots = \hat{\lambda}_k = 0$. Notice that $\operatorname{rank}(F(\hat{x})) = \hat{r} \geq s$. Following (Nikolov, 2015), we define $\hat{\Theta} := \sum_{\ell=1}^{k} \hat{\beta}_\ell \hat{u}_\ell \hat{u}_\ell^\mathsf{T}$, where

$$\hat{\beta}_\ell := \begin{cases} 1/\hat{\lambda}_\ell, & 1 \leq \ell \leq \hat{\iota}; \\ 1/\hat{\delta}, & \hat{\iota} < \ell \leq \hat{r}; \\ (1+\epsilon)/\hat{\delta}, & \hat{r} < \ell \leq k, \end{cases} \tag{3.2}$$

for any $\epsilon > 0$, where $\hat{\iota}$ is the unique integer defined in Lem. 3.6 for $\lambda_\ell = \hat{\lambda}_\ell$, and $\hat{\delta} := \frac{1}{s-\hat{\iota}} \sum_{\ell=\hat{\iota}+1}^{k} \hat{\lambda}_\ell$. From Lem. 3.6, we have that $\hat{\iota} < s$. Then,

$$-\sum_{\ell=1}^{s} \log\left(\hat{\beta}_\ell\right) = \sum_{\ell=1}^{\hat{\iota}} \log\left(\hat{\lambda}_\ell\right) + (s-\hat{\iota}) \log(\hat{\delta}) = \Gamma_s(F(\hat{x})). \tag{3.3}$$

The minimum duality gap between $\hat{x}$ in DDFact and feasible solutions of DFact of the form $(\hat{\Theta}, \upsilon, \nu, \pi, \tau)$ is the optimal value of

$$\begin{array}{l} \min\ \nu^{\mathsf{T}}\mathbf{e} + \pi^{\mathsf{T}} b + \tau s - s \\ \text{subject to:} \\ \quad \upsilon - \nu - A^{\mathsf{T}}\pi - \tau\mathbf{e} = -\operatorname{diag}(F\hat{\Theta}F^{\mathsf{T}}), \\ \quad \upsilon \geq 0,\ \nu \geq 0,\ \pi \geq 0. \end{array} \tag{$G(\hat{\Theta})$}$$

Note that $G(\hat{\Theta})$ is always feasible (e.g., $\upsilon := 0$, $\nu := \operatorname{diag}(F\hat{\Theta}F^{\mathsf{T}})$, $\pi := 0$, $\tau := 0$). Also, $G(\hat{\Theta})$ has a closed-form solution for MESP, that is with no side constraints (see (Li and Xie, 2023)).

Next, we restrict our attention to MESP, and we consider the behavior of the optimal value of $G(\hat{\Theta})$ as a function of ϵ. Let $x^* \in \{0,1\}^n$ be the support vector of the s greatest elements of $\operatorname{diag}(F\hat{\Theta}F^{\mathsf{T}})$. Then the optimal value of $G(\hat{\Theta})$ (and its dual) is $\operatorname{diag}(F\hat{\Theta}F^{\mathsf{T}})^{\mathsf{T}}x^* - s = \sum_{\ell=1}^{k} \operatorname{diag}((F\hat{U})^{\mathsf{T}} \operatorname{Diag}(x^*)(F\hat{U}))_\ell \hat{\beta}_\ell - s$, where $\hat{U}$ is the matrix having ℓ-th column equal to $\hat{u}_\ell$, for $1 \leq \ell \leq k$. It is easy to see that the diagonal elements of $(F\hat{U})^{\mathsf{T}} \operatorname{Diag}(x^*)(F\hat{U})$ are nonnegative. Therefore, with x^* fixed, $\sum_{\ell=1}^{k} \operatorname{diag}((F\hat{U})^{\mathsf{T}} \operatorname{Diag}(x^*)(F\hat{U}))_\ell \hat{\beta}_\ell$ is non-decreasing in ϵ. Now the optimal value of the dual of $G(\hat{\Theta})$, is the point-wise max, over the choices of $x^* \in \{0,1\}^n$ satisfying $\mathbf{e}^{\mathsf{T}}x^* = s$. So, the optimal value of the dual of $G(\hat{\Theta})$ is the point-wise max of linear functions, each of which is non-decreasing in ϵ. And so the optimal value of the dual is also non-decreasing in ϵ.

For developing a reasonable nonlinear-programming algorithm for DDFact, we need an expression for the gradient of its objective function.

Theorem 3.7. *Let $F(\hat{x}) = \sum_{\ell=1}^{k} \hat{\lambda}_\ell \hat{u}_\ell \hat{u}_\ell^{\mathsf{T}}$ be a spectral decomposition of $F(\hat{x})$. Let $\hat{\iota}$ be the value of ι in Lem. 3.6, where λ in Lem. 3.6 is $\hat{\lambda} := \lambda(F(\hat{x}))$. If $\frac{1}{s-\hat{\iota}} \sum_{\ell=\hat{\iota}+1}^{k} \hat{\lambda}_\ell > \hat{\lambda}_{\hat{\iota}+1}$, then, for $j = 1, 2, \ldots, n$,*

$$\frac{\partial}{\partial x_j} \Gamma_s(F(\hat{x})) = \sum_{\ell=1}^{\hat{\iota}} \frac{1}{\hat{\lambda}_\ell} (F_{j\cdot}\hat{u}_\ell)^2 + \sum_{\ell=\hat{\iota}+1}^{k} \frac{s-\hat{\iota}}{\sum_{i=\hat{\iota}+1}^{k} \hat{\lambda}_i} (F_{j\cdot}\hat{u}_\ell)^2 \ .$$

Proof. Under the hypothesis $\frac{1}{s-\hat{\iota}} \sum_{\ell=\hat{\iota}+1}^{k} \hat{\lambda}_\ell > \hat{\lambda}_{\hat{\iota}+1}$, in an open neighborhood of $\hat{\lambda}$, the value of $\hat{\iota}$ is constant. We can further check that $\hat{\lambda}_{\hat{\iota}} > \hat{\lambda}_{\hat{\iota}+1}$. Therefore, at the associated $\hat{x}$, we can

employ (Tsing, Fan, and Verriest, 1994, Thm. 3.1), and we calculate

$$\tfrac{\partial}{\partial x_j}\Gamma_s(F(\hat{x})) = \sum_{\ell=1}^{k} \tfrac{\partial \phi_s(\lambda(F(\hat{x})))}{\partial \lambda_\ell} h_j^\ell(\hat{x}) \ ,$$

where

$$h_j^\ell(\hat{x}) = \hat{u}_\ell^\mathsf{T} \frac{\partial F(\hat{x})}{\partial x_j} \hat{u}_\ell = \hat{u}_\ell^\mathsf{T} F_{j\cdot}^\mathsf{T} F_{j\cdot} \hat{u}_\ell = (F_{j\cdot}\hat{u}_\ell)^2 \ .$$

Calculating

$$\frac{\partial \phi_s(\hat{\lambda})}{\partial \lambda_\ell} = \begin{cases} 1/\hat{\lambda}_\ell \ , & \text{if } \ell \leq \hat{\iota}; \\ \frac{s-\hat{\iota}}{\sum_{i=\hat{\iota}+1}^{k} \hat{\lambda}_i}, & \text{if } \ell > \hat{\iota}, \end{cases}$$

the result follows. □

Without the technical condition $\frac{1}{s-\hat{\iota}} \sum_{\ell=\hat{\iota}+1}^{k} \hat{\lambda}_\ell > \hat{\lambda}_{\hat{\iota}+1}$, the formulae above still give a subgradient of Γ_s (see (Li and Xie, 2023) for details).

Finally, considering DDFact, we can now present a short proof for Cor. 3.3.

Proof [Cor. 3.3]. As $z_{\text{DDFact}}(C, s, A, b; F) = z_{\text{DFact}}(C, s, A, b; F)$, it suffices to show that the objective value of DDFact at any feasible solution $\hat{x}$, does not dependent on the factorization $C = FF^\mathsf{T}$. We have that $F(\hat{x}) := F^\mathsf{T} \operatorname{Diag}(\hat{x}) F$ has the same non-zero eigenvalues as $(\operatorname{Diag}(\hat{x}))^{\frac{1}{2}} FF^\mathsf{T} (\operatorname{Diag}(\hat{x}))^{\frac{1}{2}} = (\operatorname{Diag}(\hat{x}))^{\frac{1}{2}} C (\operatorname{Diag}(\hat{x}))^{\frac{1}{2}}$. The result follows.

3.2.4 linx

For $x \in [0,1]^n$ and $\gamma > 0$, we define $K_\gamma(x) := \gamma C \operatorname{Diag}(x) C + \operatorname{Diag}(\mathbf{e} - x)$ in linx.

It turns out that the linx bound is invariant under complementing ((Anstreicher, 2020); also see (Fampa and Lee, 2022)), while the factorization bound is not; therefore, we can obtain a different bound value for CMESP by considering the factorization bound on the complementary problem.

We can give an expression for the gradient and the Hessian of the linx objective function. Using well-known facts, we can work out that for all $\hat{x}$ in the domain of the objective function, we have $\nabla \left(\frac{1}{2} \operatorname{ldet} K_\gamma(\hat{x})\right) = \frac{1}{2} \left(\operatorname{diag}(\gamma C K_\gamma(\hat{x})^{-1} C) - \operatorname{diag}(K_\gamma(\hat{x})^{-1})\right)$, and $\nabla^2 \left(\frac{1}{2} \operatorname{ldet} K_\gamma(\hat{x})\right) =$

$$\tfrac{1}{2} \Bigg(- \gamma^2 (CK_\gamma(\hat{x})^{-1}C) \circ (CK_\gamma(\hat{x})^{-1}C) \\ + \gamma \Big((K_\gamma(\hat{x})^{-1}C) \circ (K_\gamma(\hat{x})^{-1}C) + ((K_\gamma(\hat{x})^{-1}C) \circ (K_\gamma(\hat{x})^{-1}C))^\mathsf{T} \Big) - K_\gamma(\hat{x})^{-1} \circ K_\gamma(\hat{x})^{-1} \Bigg).$$

3.2.5 Mixing

We consider $m \geq 1$ convex relaxations for CMESP, indexed by $i = 1, \ldots, m$:

$$v_i := \max \left\{ f_i(L_i(x)) \ : \ \mathbf{e}^\mathsf{T} x = s, \ Ax \leq b, \ 0 \leq x \leq \mathbf{e} \right\},$$

where, for $i = 1, \ldots, m$, $k_i \leq n$, $L_i : \mathbb{R}^n \to \mathbb{S}^{k_i}_+$ are affine functions, and $f_i : \mathbb{S}^{k_i}_+ \to \mathbb{R}$ are concave functions. We write $L_i(x) := L_{i0} + L_{i1}x_1 + \cdots + L_{in}x_n$ and $L_{ij} \in \mathbb{S}^{k_i}$, for $i = 1, \ldots, m$ and $j = 0, \ldots, n$, and we note that the objective functions of DDFact, complementary DDFact, and linx can be written as $f_i(L_i(x))$ (see §3.2.5.1).

For a "weight vector" $\alpha \in \mathbb{R}^m_+$, such that $\mathbf{e}^\mathsf{T}\alpha = 1$, we define the *mixing bound* (see (Chen, Fampa, Lambert, and Lee, 2021) for a more general setting):

$$v(\alpha) := \max \left\{ \textstyle\sum_{i=1}^m \alpha_i f_i(L_i(x)) \ : \ \mathbf{e}^\mathsf{T} x = s, \ Ax \leq b, \ 0 \leq x \leq \mathbf{e} \right\}. \tag{mix}$$

The goal is to minimize the mixing bound over α (and any parameters for the bounds).

We construct the Lagrangian dual of mix for a broad class of cases that covers our applications of mixing. For $i = 1, \ldots, m$, we assume that for any given $\hat{\Theta}_i \in \mathbb{S}^{k_i}_{++}$, there is a closed-form solution $\hat{W}_i$ to $\sup\{f_i(W_i) - \hat{\Theta}_i \bullet W_i \ : \ W_i \succeq 0\}$, such that $\hat{\Theta}_i \bullet \hat{W}_i =: \rho_i \in \mathbb{R}$ and $\Omega_i : \mathbb{S}^{k_i}_{++} \to \mathbb{R}$, is defined by $\Omega_i(\hat{\Theta}_i) := f_i(\hat{W}_i)$. Furthermore, we assume that the supremum is $+\infty$ if $\hat{\Theta}_i \not\succ 0$.

With the assumptions above, the Lagrangian dual problem of mix is equivalent to

$$\begin{array}{l} z_{\text{Dmix}}(C, s, A, b) := \min \ \ \textstyle\sum_{i=1}^m \alpha_i \Big(\Omega_i(\Theta_i) - \rho_i + \Theta_i \bullet L_{i0} \Big) + \nu^\mathsf{T}\mathbf{e} + \pi^\mathsf{T} b + \tau s \\ \quad \text{subject to:} \\ \quad \textstyle\sum_{i=1}^m \alpha_i \Big(\Theta_i \bullet L_{ij} \Big) + \upsilon_j - \nu_j - \pi^\mathsf{T} A_{\cdot j} - \tau = 0, \quad \text{for } j \in N, \\ \quad \Theta_1 \succ 0, \ldots, \Theta_m \succ 0, \ \upsilon \geq 0, \ \nu \geq 0, \ \pi \geq 0. \end{array} \tag{Dmix}$$

Theorem 3.8. *Let*

- *LB be the objective-function value of a feasible solution for CMESP,*
- $(\hat{\Theta}_1, \ldots, \hat{\Theta}_m, \hat{\upsilon}, \hat{\nu}, \hat{\pi}, \hat{\tau})$ *be a feasible solution for Dmix with objective-function value* $\hat{\zeta}$.

Then, for every optimal solution x^* *for CMESP, we have:*

$$\begin{array}{l} x^*_j = 0, \ \forall \ j \in N \text{ such that } \hat{\zeta} - LB < \hat{\upsilon}_j \ , \\ x^*_j = 1, \ \forall \ j \in N \text{ such that } \hat{\zeta} - LB < \hat{\nu}_j \ . \end{array}$$

Proof. Analogous to the proof of Thm. 3.5. □

Next, we generalize to Dmix, the procedure presented in §3.2.3 to construct a feasible solution of DFact from a feasible solution of DDFact. A good feasible solution for Dmix can be used to validate the quality of the solution obtained for mix, and to fix variables by applying the result of Thm. 3.8.

We let $\hat{x}$ be a feasible solution of mix in the domain of f_i and define $\hat{W}_i := L_i(\hat{x})$, for $i = 1, \ldots, m$. First, we assume that it is possible to compute $\hat{\Theta}_i$, such that $\Omega_i(\hat{\Theta}_i) = f_i(\hat{W}_i)$, for $i = 1, \ldots, m$. Then, the minimum duality gap between $\hat{x}$ in mix and feasible solutions of Dmix of the form $(\hat{\Theta}_1, \ldots, \hat{\Theta}_m, \upsilon, \nu, \pi, \tau)$ is the optimal value of the linear program

$$
\begin{array}{l}
\min\ \nu^{\mathsf{T}}\mathbf{e} + \pi^{\mathsf{T}} b + \tau s - \sum_{i=1}^{m} \alpha_i \Big(\rho_i - \hat{\Theta}_i \bullet L_{i0}\Big) \\
\quad \text{subject to:} \\
\quad \upsilon_j - \nu_j - \pi^{\mathsf{T}} A_{\cdot j} - \tau = -\sum_{i=1}^{m} \alpha_i \left(\hat{\Theta}_i \bullet L_{ij}\right), \quad \text{for } j \in N, \\
\quad \upsilon \geq 0,\ \nu \geq 0,\ \pi \geq 0.
\end{array}
\qquad (G(\hat{\Theta}_1, \ldots, \hat{\Theta}_m))
$$

Analogously to $G(\hat{\Theta})$, we can verify that $G(\hat{\Theta}_1, \ldots, \hat{\Theta}_m)$ is always feasible and has a simple closed-form solution for MESP; the only differences between $G(\hat{\Theta})$ and $G(\hat{\Theta}_1, \ldots, \hat{\Theta}_m)$ are the constant in the objective function and the right-hand side of the constraints.

3.2.5.1 Considering DDFact, complementary DDFact, and linx in mix

Considering $f_i(L_i(x))$ as the objective function of DDFact we have $f_i(\cdot) := \Gamma_s(\cdot)$ and $L_i(x) := F^{\mathsf{T}} \operatorname{Diag}(x) F$, so $k_i := k$, $L_{i0} := 0$ and $L_{ij} := F_{j\cdot}^{\mathsf{T}} F_{j\cdot}$, for $j = 1, \ldots, n$. We also have $\rho_i := s$ and $\Omega_i(\Theta_i) := -\sum_{\ell=k-s+1}^{k} \log(\lambda_\ell(\Theta_i))$. For a given feasible solution $\hat{x}$ of mix in the domain of f_i and $\hat{W}_i := L_i(\hat{x})$, construct $\hat{\Theta}_i$ as discussed in §3.2.3, and we see in (3.3), that $\Omega_i(\hat{\Theta}_i) = f_i(\hat{W}_i)$.

Considering $f_i(L_i(x))$ as the objective function of complementary DDFact we have $f_i(\cdot) := \Gamma_{n-s}(\cdot) + \operatorname{ldet} C$ and $L_i(x) := F^{-1} \operatorname{Diag}(\mathbf{e} - x) F^{-\mathsf{T}}$, so $k_i := k(= n)$, $L_{i0} := F^{-1} F^{-\mathsf{T}}$ and $L_{ij} := -F_{j\cdot}^{-1} F_{j\cdot}^{-\mathsf{T}}$, for $j = 1, \ldots, n$. We also have $\rho_i := n - s$ and $\Omega_i(\Theta_i) := -\sum_{\ell=k-n+s+1}^{k} \log(\lambda_\ell(\Theta_i)) + \operatorname{ldet} C$. For a given feasible solution $\hat{x}$ of mix in the domain of f_i and $\hat{W}_i := L_i(\hat{x})$, construct $\hat{\Theta}_i$ as discussed in §3.2.3, and we see in (3.3), that $\Omega_i(\hat{\Theta}_i) = f_i(\hat{W}_i)$.

Considering $f_i(L_i(x))$ as the objective of linx we have $f_i(\cdot) := \frac{1}{2}(\operatorname{ldet}(\cdot) - s \log \gamma)$ and $L_i(x) := \gamma C \operatorname{Diag}(x) C + \operatorname{Diag}(\mathbf{e} - x)$, so $k_i := n$, $L_{i0} := I$ and $L_{ij} := \gamma C_{j\cdot}^{\mathsf{T}} C_{j\cdot} - \mathbf{e}_j \mathbf{e}_j^{\mathsf{T}}$, for $j = 1, \ldots, n$. We have $\rho_i := n/2$ and $\Omega_i(\Theta_i) := -\frac{1}{2}(\operatorname{ldet}(2\Theta_i) + s \log \gamma)$. For a feasible solution $\hat{x}$ of mix in the domain of f_i and $\hat{W}_i := L_i(\hat{x})$, set $\hat{\Theta}_i := \frac{1}{2}\hat{W}_i^{-1}$, and

then $\Omega_i(\hat{\Theta}_i) = f_i(\hat{W}_i)$.

We note that if the objective of mix is a weighted combination of the three functions mentioned above and $\hat{x}$ is an optimal solution to mix, then the optimal objective value of $G(\hat{\Theta}_1, \ldots, \hat{\Theta}_m)$ is zero, that is, the dual solution constructed to Dmix is also optimal.

3.3 Implementation and experiments

3.3.1 Setup for the computational experiments

(Li and Xie, 2023) worked with solving DDFact with respect to MESP, using a custom-built Frank-Wolf (see (Frank and Wolfe, 1956)) style code, written in Python. They only worked with the relaxation, and did not seek to solve MESP to optimality. The linx bound for CMESP was introduced by (Anstreicher, 2020), where bound calculations were carried out with the conical-optimization software `SDPT3` (see (Toh, Todd, and Tütüncü, 1999)), within the very-convenient `Yalmip` `Matlab` framework (see (Lofberg, 2004)), and a full branch-and-bound code for MESP was written in `Matlab`.

In our experiments, we calculate all of our bounds using a single state-of-the-art commercial nonlinear-programming solver, to facilitate fair comparisons between bounding methods, and also to see what is possible in such a computational setting.

We experimented on instances of MESP and CMESP with linx, DDFact and complementary DDFact (i.e, DDFact applied to CMESP-comp). We ran our experiments under Windows, on an Intel Xeon E5-2667 v4 @ 3.20 GHz processor equipped with 8 physical cores (16 virtual cores) and 128 GB of RAM. We implemented our code in `Matlab` using the commercial software `Knitro`, version 12.4, as our nonlinear-programming solver. `Knitro` offers BFGS-based algorithms and Newton-based algorithms to solve nonlinear programs. In the first case, `Knitro` only needs function values and gradients from the user, in the latter, `Knitro` also needs second derivatives. By experimenting on top of one state-of-the-art general-purpose nonlinear-programming code, we hoped to get good and rapid convergence and get running times that can reasonably be compared for the different relaxations. In all of our experiments we set `Knitro` parameters[1] as follows: `algorithm` $= 3$ to use an active-set method, `convex` $= 1$ (true), `gradopt` $= 1$ (we provided exact gradients), `maxit` $= 1000$. We set `opttol` $= 10^{-10}$, aiming to satisfy the KKT optimality conditions to a very tight tolerance. We set `xtol` $= 10^{-15}$ (relative tolerance for lack of progress in the solution point) and `feastol` $= 10^{-10}$ (relative tolerance for the feasibility error), aiming for the best solutions that we could reasonably find.

[1] see https://www.artelys.com/docs/knitro/2_userGuide.html, for details

3.3.2 Test instances

To compare the bounds obtained with the three relaxations, we consider four covariance matrices from the literature, with $n = 63, 90, 124, 2000$. For each matrix, we consider different values of s defining a set of test instances of MESP. The $n = 63$ and $n = 124$ matrices are benchmark covariance matrices obtained from J. Zidek (University of British Columbia), coming from an application to re-designing an environmental monitoring network; see (Guttorp, Le, Sampson, and Zidek, 1993) and (Hoffman, Lee, and Williams, 2001). The $n = 90$ matrix is based on temperature data from monitoring stations in the Pacific Northwest of the United States; see (Anstreicher, 2020). These $n = 63, 90, 124$ matrices are all nonsingular. All of these matrices have been used extensively in testing and developing algorithms for MESP; see (Ko, Lee, and Queyranne, 1995; Lee, 1998; Anstreicher, Fampa, Lee, and Williams, 1999; Lee and Williams, 2003; Hoffman, Lee, and Williams, 2001; Anstreicher and Lee, 2004; Burer and Lee, 2007; Anstreicher, 2018, 2020). The largest covariance matrix that we considered in our experiments is an $n = 2000$ matrix with rank 949, based on Reddit data, used in (Li and Xie, 2023) and from (Dey, Mazumder, and Wang, 2022) (also see (Bagroy, Kumaraguru, and De Choudhury, 2017)). To ameliorate some instability in running times, for $n = 63, 90, 124$ we repeated every experiment ten times, and for $n = 2000$, we repeated every experiment five times, and present average timing results.

3.3.3 Numerical experiments for $n = 63, 90, 124$

For the three nonsingular covariance matrices used in our experiments, we solved linx, DDFact and complementary DDFact, for all $2 \leq s \leq n - 1$. For each matrix, we present four plots. In the first plots of Figures 3.1, 3.2 and 3.3, we present the integrality gap for each bound and each s. Each such gap is given by the difference between the upper bound computed by solving the relaxation and a lower bound obtained using a heuristic of (Lee, 1998, §4) followed by a simple local search (see (Ko, Lee, and Queyranne, 1995, §4)).

In the second plots of those figures, we present the average wall-clock times (in seconds) used by `Knitro` to solve the relaxations. Some observations about the times presented are important. First, we note that the times depicted on the plots correspond to the application of a BFGS-based algorithm to solve DDFact and complementary DDFact, and to the application of a Newton-based algorithm to solve linx. As an experiment, we also applied a BFGS-based algorithm to solve linx, not passing the Hessian to the solver, but, as expected, the results were worse concerning both time and convergence of the algorithm. The difference between the times can be seen in Table 3.2 (aggregated over s) and Figure 3.5. On the other hand, we did not apply a Newton-based algorithm to DDFact and complemen-

tary DDFact because we cannot guarantee that the objective function of these relaxations is differentiable at every iterate (of the nonlinear-programming solver). We should also note that the times shown in the plots for linx do not include the times to compute the value of the parameter γ in the problem formulation. This parameter value has a great impact on the linx bound. We present the times to compute them, aggregated over s, in Table 3.2. Finally, we should mention that `Knitro` did not prove optimality for several instances solved. However, we could confirm the optimality of *all* solutions returned by `Knitro`, up to the optimality tolerance considered, by constructing a dual solution with duality gap less than the tolerance, with respect to the primal solution obtained by `Knitro`. To construct the dual solutions, we solved various special cases of the linear program $G(\hat{\Theta}_1, \ldots, \hat{\Theta}_m)$ (see §3.2.5). For DDFact, the construction uses (3.2), where we took $\epsilon = 0$, which gives us a dual solution that is feasible within numerical accuracy.

On our experiments with instances of MESP (i.e., no side constraints), these linear programs have closed-form solutions and the times to compute them are not significant.

In the third plots of Figures 3.1, 3.2 and 3.3, we demonstrate the capacity of the mixing methodology described in §3.2.5 to decrease the integrality gap. As observed in (Chen, Fampa, Lambert, and Lee, 2021), the methodology is particularly effective when considering in mix, a weighted sum of the objective functions of two relaxations, such that the bounds obtained by each relaxation are close to each other. We exploit this observation in our experiments. For each covariance matrix, we select one or more pairs of relaxations for which the integrality-gap curves (presented in the first plots of the figures) cross each other at some point. Then, we mix these two relaxations and compute new mixed bounds for all values of s in a promising interval, approximately centered at the point where the two curves cross. To select the parameter α that weights the objective in mix, we simply apply a bisection algorithm.

Finally, in the fourth plots of Figures 3.1, 3.2, and 3.3, we demonstrate how effective the strategy described in Thm.s 3.5 and 3.8 can be to fix variables (e.g., the context could be fixing variables at the root node of the enumeration tree in applying a branch-and-bound algorithm). In all of our experiments, we use a fixing threshold of 10^{-10} which can be considered as rather safe in the context of the accuracy that we use to compute the relevant quantities. Although the mixing strategy can decrease the integrality gap for some instances, in our experiments this improvement is not enough to allow more variables to be fixed. Therefore, we do not consider the mixed bounds in these plots.

3.3.4 Analysis of the results for $n = 63, 90, 124$

The analysis of the plots for the $n = 63$ and $n = 90$ covariance matrices are very similar to each other and are summarized in the following.

- We see from the first plots of Figures 3.1 and 3.2 that the complementary DDFact bound is not competitive with the DDFact and linx bounds for these instances. For most values of s the first bound is much worse than the two others. The complementary DDFact bound is only a bit better than the DDFact bound for very large values of s and it is never better than the linx bound. The integrality-gap curves for DDFact and linx cross at points close to an intermediate value of s. For smaller s, DDFact gives the best bound and for larger s, linx is the winner.

- Concerning the wall-clock time to compute the bounds, we see again a big disadvantage of complementary DDFact in the second plots of Figures 3.1 and 3.2. Although it is faster than DDFact on some instances with large s, we see that on most instances its time is much longer than the times for the two other relaxations, and with the greatest variability among the different values of s. The solution of linx is always significantly faster than the solution of the two other relaxations for these instances. Finally, we note that the variation in the time to solve linx for all values of s is less than 1%, while there is a great variability for the other two relaxations.

- We see in the first plots of Figures 3.1 and 3.2 that the curves corresponding to DDFact and linx cross at points close to intermediate values of s, indicating a promising interval of values to mix these relaxations for both $n = 63$ and $n = 90$. Considering such intervals, we see in the third plots of Figures 3.1 and 3.2, how mixing DDFact and linx can in fact, decrease the integrality gap for some instances, being mostly effective for the values of s for which the DDFact bound and the linx bound are very close to each other.

- In the fourth plots of Figures 3.1 and 3.2 we verify the increasing capacity to fix variables as the bound gets stronger. Interestingly, we see that for large values of s, complementary DDFact bounds can lead to more variables fixed than the better DDFact bound.

For $n = 124$, we have a slightly different analysis because, as we see in the first plot of Figure 3.3, the complementary DDFact bound becomes better than the DDFact bound for all s larger than an intermediate value. We note that we can observe this same behavior with the "NLP" relaxation for CMESP used in (Anstreicher, Fampa, Lee, and Williams,

1999). Moreover, we see in the first plot of Figure 3.3, two points where the curves cross, showing three interesting intervals for s, where each one of the three relaxations gives the best bound. Concerning the wall-clock time, the observations about the second plot of Figure 3.3 are similar to the ones about the second plots of Figures 3.1 and 3.2, confirming that the time to solve linx is shorter and with a smaller variability with s, when compared to the two other relaxations. In the third plot of Figure 3.3, we exploit the three crossing points of the integrality-gap curves for $n = 124$, and show separately the capacity of the mixing methodology to decrease the gaps when we mix the two relaxations corresponding to each crossing point. It is interesting to note that the mixing methodology is more effective when the crossing curves are less flat at those points, that is, when the gaps change faster as s changes. Finally, we have different observations about the fourth plot of Figure 3.3 when compared to the smaller instances, concerning the capacity of the relaxations to fix variables. As DDFact and complementary DDFact lead to very small integrality gaps at both ends of the curves, we observe in the fourth plot of Figure 3.3 their stronger capacity of fixing variables on the corresponding values of s, when compared to linx. For $n = 124$, we see that linx gives the best bounds for intermediate values of s only. These are clearly the most difficult instances for $n = 124$. Therefore, the integrality gap is usually not small enough on these instances to allow variable fixing.

3.3.5 Numerical experiments with the large instance ($n = 2000$)

The bounds computed for the $n = 2000$ matrix are analyzed in Figure 3.4. As this larger matrix is singular, we could not apply the complementary DDFact relaxation to obtain a bound. In the first plot, we present the integrality gaps for DDFact and linx for all $20 \leq s \leq 200$ that are multiples of 20. For lower bounds in computing the gaps, we obtained them by the same heuristic applied to our smaller instances with $n = 63, 90, 124$. We clearly see the superiority of DDFact for this input matrix, for these relatively small values of s, following the behavior observed for the smaller instances. Concerning the wall-clock time, we still see in the second plot that linx can be solved faster on the most difficult instances with $s \geq 100$, and once more, we see a very small variability in the times for linx, unlike what we see for DDFact. The significant rank deficiency of the covariance matrix of these instances would seem to be a disadvantage for linx relative to DDFact, with regard to the computational time needed to solve them; this is because the order of the matrix considered in the objective function of linx is always equal to the order of the covariance matrix, while for DDFact it is given by its rank. As linx could not fix variables for any value of s, we present in the third plot only the number of variables fixed for each s, considering the DDFact bound,

and we can see that the fixing procedure is very effective when $s \leq 80$.

Our success with fixing using the DDFact bound on the $n = 2000$ matrix, for $s = 20, 40, 60, 80$, gave us some hope to solve these instances to optimality, or at least reduce them to a size where we could realistically hope that branch-and-bound could succeed. So we devised an iterative fixing scheme, applying fixing to a sequence of reduced instances, with the goal of solving to optimality or at least fixing substantially more variables. At each iteration, we calculated and attempted to fix based on the DDFact bound *and* the linx bound. We re-applied the heuristic for a reduced problem, in case it could improve on the lower bound of its parent. Even though the linx bound cannot fix any variables at the first iteration, for $s = 20, 40, 60$, it enabled us to fix more variables for reduced problems. For $s = 20, 40, 60$, we could solve to optimality. The results are summarized in Table 3.1, where s' and n' are the parameters for reduced problems, by iteration, and $*$ indicates the iteration where we can assert that fixing identified the optimal solution. Unfortunately, for $s = 80$, linx could not fix anything after one round of DDFact fixing. We noted that for $s = 20, 40, 60$, the heuristic applied to the $n = 2000$ root instance gave what turned out to be the optimal solution. It is possible that we did not succeed on $s = 80$ because our lower bound is not strong enough.

We carried out some additional experiments for the $n = 2000$ matrix, with $s = 860, 880, \ldots, 940$ (recall that the rank of C is 949). For these instances, DDFact is very hard for `Knitro` to solve: the solution times for DDFact are an order of magnitude larger as compared to the instances with $20 \leq s \leq 200$; this is not the case for linx. Additionally, `Knitro` failed to converge for $s = 920$. In any case, for all of these problems that we could solve, we had huge integrality gaps, and no variables could be fixed based on either linx or DDFact.

3.3.6 More specifics about the computational time

In Table 3.2, we show means and standard deviations of the average wall-clock times (in seconds) for the main procedures considered in our experiments and for each n. For $n = 63, 90, 124$, the statistics consider the solution for all $2 \leq s \leq n - 1$. For $n = 2000$, the statistics consider all $20 \leq s \leq 200$ that are multiples of 20.

In Table 3.2, the columns "DDFact", "DDFactcomp" and "linx (Newton)" summarize information depicted in the second plots of Figures 3.1, 3.2, and 3.3. For each n, the mean and standard deviation of the times increase in the order: linx, DDFact, complementary DDFact, and the means and standard deviations are all significantly better for linx.

In Table 3.2, the column "linx (BFGS)" presents the mean and standard deviation of

the times (across all s) when linx is solved by `Knitro` without passing the Hessian of the objective function to the solver, i.e., with the application of a BFGS-based algorithm. We see that not passing the Hessian of the objective function to the solver leads to a significant increase in the solution time, and also in the variability of the times across the different values of s. Although we can see performance aggregated over s in the "linx (Newton)" and "linx (BFGS)" columns, in Figure 3.5, we get a more complete view of the strong dominance, across most s for each input matrix C. We have plotted the time for linx (Newton) divided by time for linx (BFGS), against s/n. With the vast majority of the ratios being less than one for each input matrix, and this emphatically being the case for the $n = 2000$ matrix, we can confidently recommend passing the Hessian to `Knitro` when solving linx.

In Table 3.2, the column "γ (Newton)" in Table 3.2 presents statistics for the time used to compute the value of the parameter γ used in linx across the different s. To optimize γ we do a one-dimensional search, exploiting the fact that the linx bound is convex in the logarithm of γ (see (Chen, Fampa, Lambert, and Lee, 2021)), and we use `Knitro` passing the Hessian at each iteration of the one-dimensional search. We observe that, compared to solving linx, optimizing γ is very expensive, however, we should note that the optimization procedure applied had no concern with time. When time is relevant, as in the context of a branch-and-bound algorithm, we can apply a faster procedure, like the one applied in (Anstreicher, 2020). Furthermore, we should notice that in a branch-and-bound context, the linx bound is computed for each subproblem considered, but the parameter γ should not be optimized for every one of them. As done in (Anstreicher, 2020), it would be more efficient to use the same parameter value as the one used on the parent node most of the times.

3.3.7 Some experiments with CMESP

To illustrate the application of our bounds to CMESP, we repeated the experiments performed with the instances of MESP with covariance matrix of dimension $n = 63$ and $5 \leq s \leq 47$, but now including five side constraints $a_i^T x \leq b_i$, for $i = 1, \ldots, 5$. The left-hand side of constraint i is given by a uniformly-distributed random vector a_i with integer components between 1 and 5. The right-hand side of the constraints was selected so that, for every $5 \leq s \leq 47$, the best known solution $x^*(s)$ of the instance of MESP is violated by at least one constraint. For that, each b_i was selected as the 80-th percentile of the values $a_i^T x^*(s) - 1$, for all $5 \leq s \leq 47$. We note that when considering side constraints, the linear program $G(\hat{\Theta}_1, \ldots, \hat{\Theta}_m)$ does not have a closed-form solution. In this case, we solve it with `Knitro`. The time needed to calculate the dual solution with the `Knitro` linear-programming solver is no more than 5% of the time needed to calculate the DDFact bound or the complementary

DDFact bound. However, for the linx bound, the variability of the times is large; for some instances, the time needed to construct the dual solution can even exceed the time needed to calculate the linx bound.

In Figure 3.6 we present plots for CMESP, analogous to those shown in Figure 3.1 for MESP, considering our instances of dimension $n = 63$. We have a very similar analysis of the results presented in both figures, illustrating the robustness of our approach when including side constraints to MESP.

3.4 Concluding remarks

We developed useful properties of the DDFact bound, aimed at guiding computational practice. In particular, we saw that (i) the DDFact bound is invariant under the factorization of the input matrix C, (ii) the DDFact bound cannot be improved by scaling C, and (iii) the DDFact bound dominates the spectral bound. We developed a fixing scheme for DDFact, we showed how to mix DDFact with linx and with complementary DDFact (see (Chen, Fampa, Lambert, and Lee, 2021, §7)) for general comments on how and when mixing could potentially be employed within B&B), and we gave a general variable-fixing scheme for mixings.

Overall, we found that working with a general-purpose NLP solver is quite practical for solving linx and DDFact relaxations of CMESP. For DDFact, this is despite the fact that its objective function is not guaranteed to be smooth at all iterates (of the NLP solver). We found that for linx, which has a smooth objective function, passing the Hessian to the NLP solver is quite effective; the running times are much better, in mean and variance, compared to a BFGS-based approach. We found that various mixings of linx, DDFact and complementary DDFact can lead to improved bounds. We found that fixing can be quite effective for DDFact and complementary DDFact. Unfortunately, we did not find mixing to be useful for fixing additional variables, as compared to fixing variables based on each relaxation separately, on the benchmark instances that we experimented with. But we did find iterative fixing, employing the linx and DDFact fixing rules in concert, to be quite effective on large and difficult instances; for the $n = 2000$ matrix and $s = 20, 40, 60$, we could find and verify optimal solutions for the first time, and without any branching.

In future work, we plan to develop a full B&B implementation aimed at solving difficult instances of CMESP to optimality. Our experiments on benchmark covariance matrices indicate that such an approach should use both the linx and DDFact bounds. In particular, linx often seems to be valuable for the large values of s (where DDFact deteriorates). We can hope that variable fixing can be exploited for subproblems, and we expect mixing to be most valuable for subproblems having s near the middle of the range.

Additionally, we plan on working further on improving the algorithmics for the DDFact relaxation, for modern settings in which the order n of the covariance matrix greatly exceeds its rank r. Specifically: (i) we plan to take better advantage of the fact that the matrix in the objective function of DDFact is order r while in linx it is order n; (ii) while the objective function of DDFact is not guaranteed to be smooth at all iterates, we found that it usually is, and so we plan to use second-order information to improve convergence.

Iter	s'	n'	s'	n'	s'	n'	s'	n'
0	20	2000	40	2000	60	2000	80	2000
1	20	28	40	58	60	110	80	442
2	2	7	6	13	22	68		
3	*	*	3	4	16	24		
4			1	2	1	2		
5			*	*	*	*		

Table 3.1: Iterated fixing for $n = 2000$

n	DDFact		DDFactcomp		linx (Newton)		linx (BFGS)		γ (Newton)	
	mean	std	mean	std	mean	std	mean	std	mean	std
63	0.1807	0.0828	0.2756	0.2496	0.0459	0.0061	0.0518	0.0092	0.4202	0.0832
90	0.3653	0.1327	0.4163	0.2398	0.0629	0.0094	0.0849	0.0202	0.6727	0.1376
124	0.5023	0.2373	0.7451	0.3892	0.0874	0.0142	0.1005	0.0177	1.3043	0.2675
2000	461.49	536.37	-	-	133.69	17.54	542.93	530.50	2242.90	645.37

Table 3.2: Wallclock time (sec)

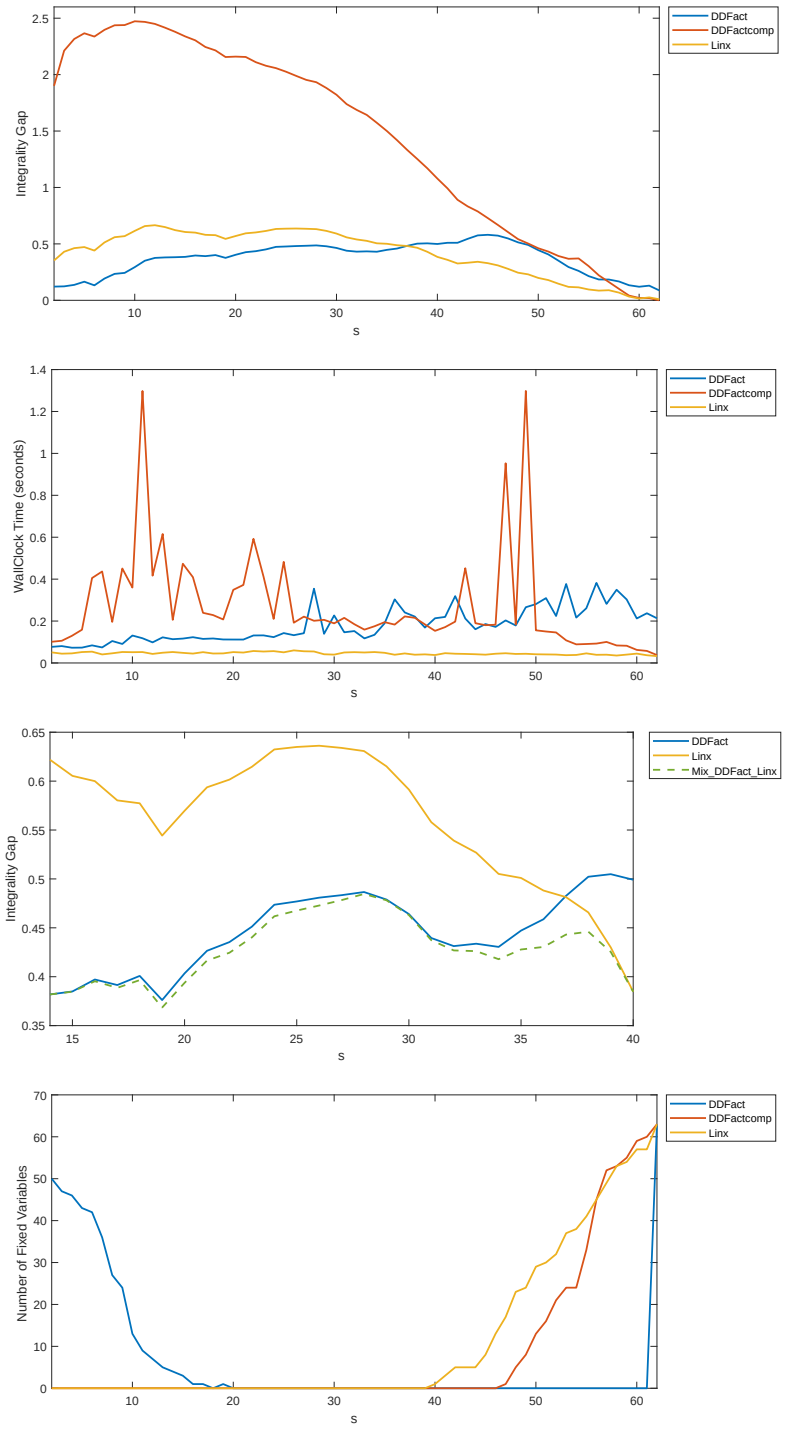

Figure 3.1: Bounds/times comparison and effect of the mixing and variable-fixing methodologies for $n = 63$

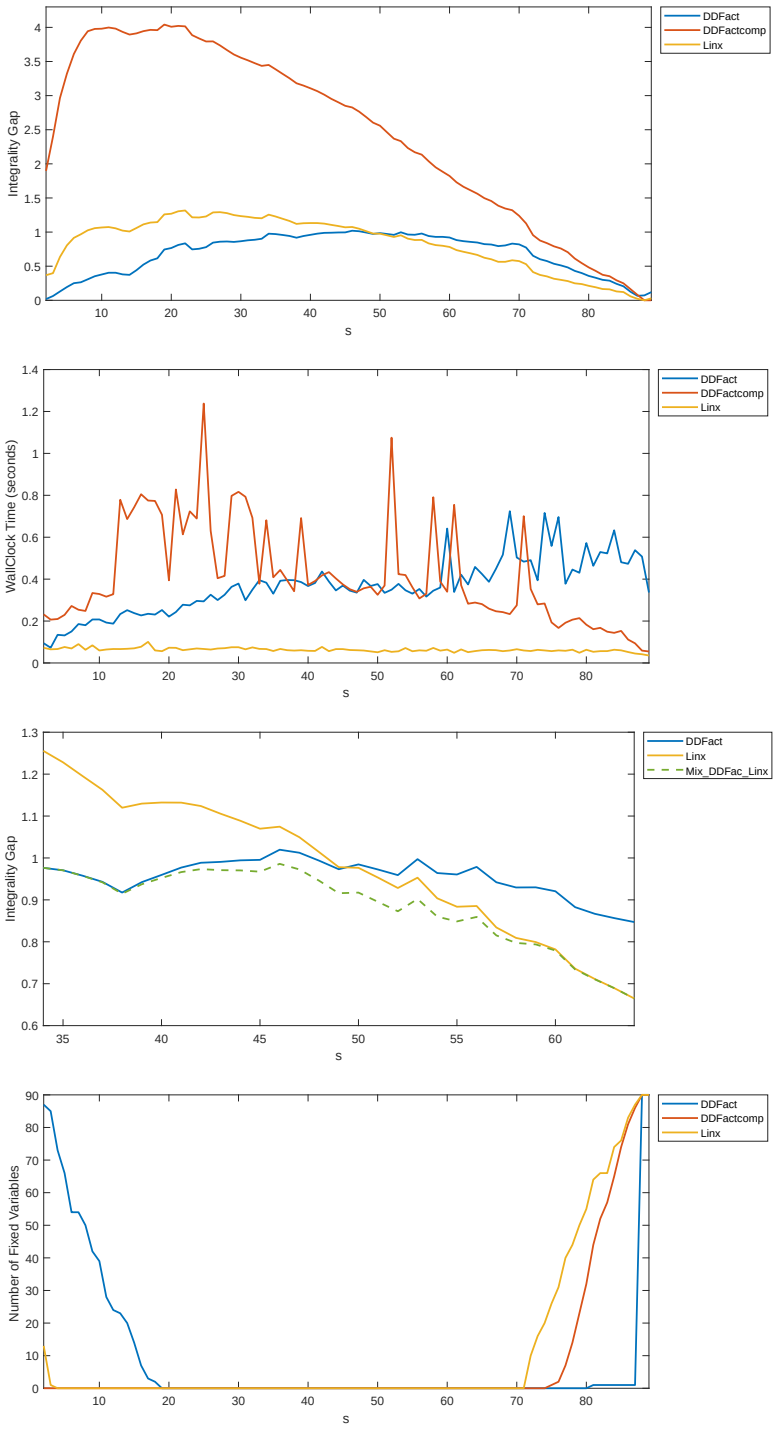

Figure 3.2: Bounds/times comparison and effect of the mixing and variable-fixing methodologies for $n = 90$

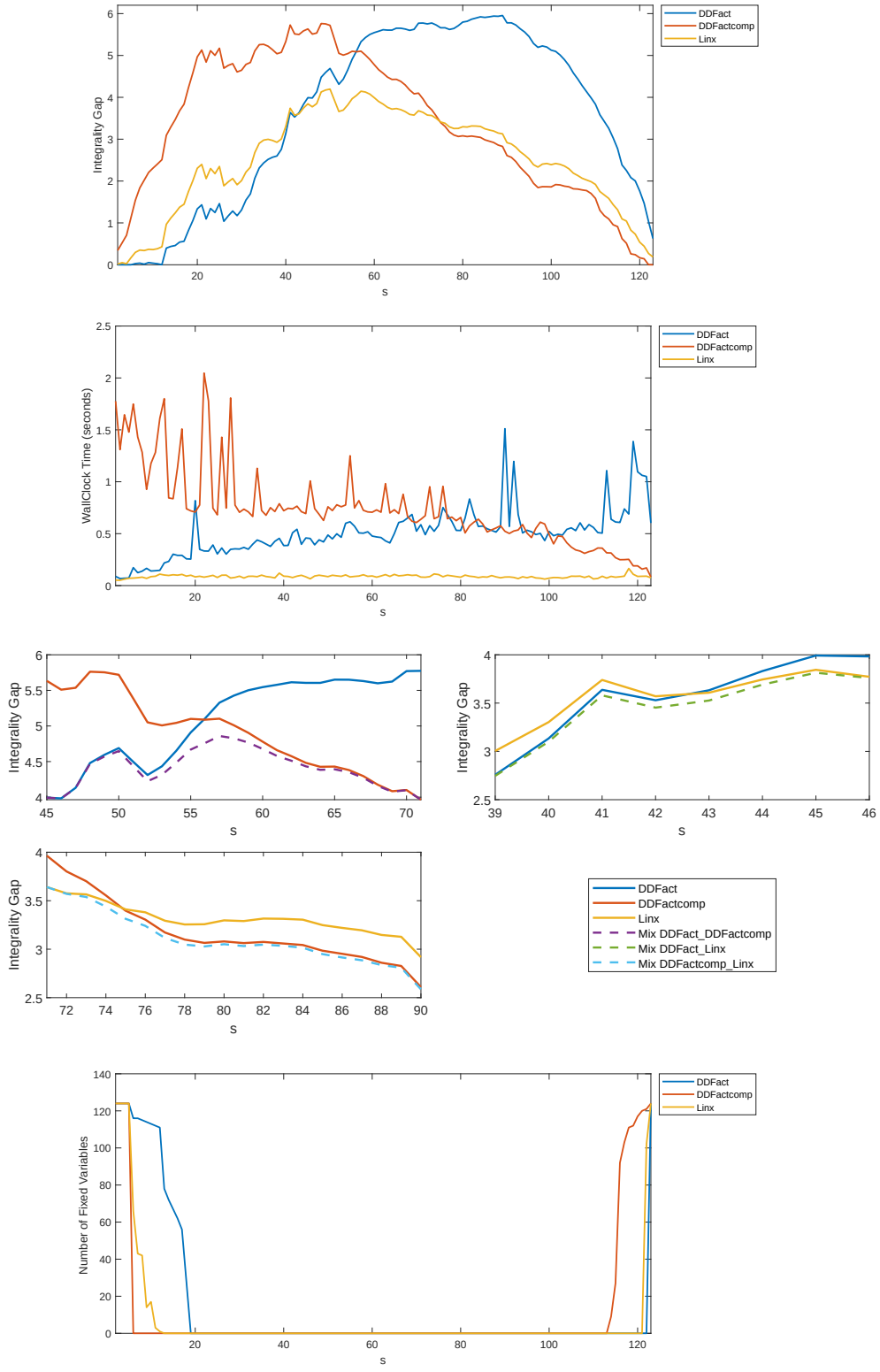

Figure 3.3: Bounds/times comparison and effect of the mixing and variable-fixing methodologies for $n = 124$

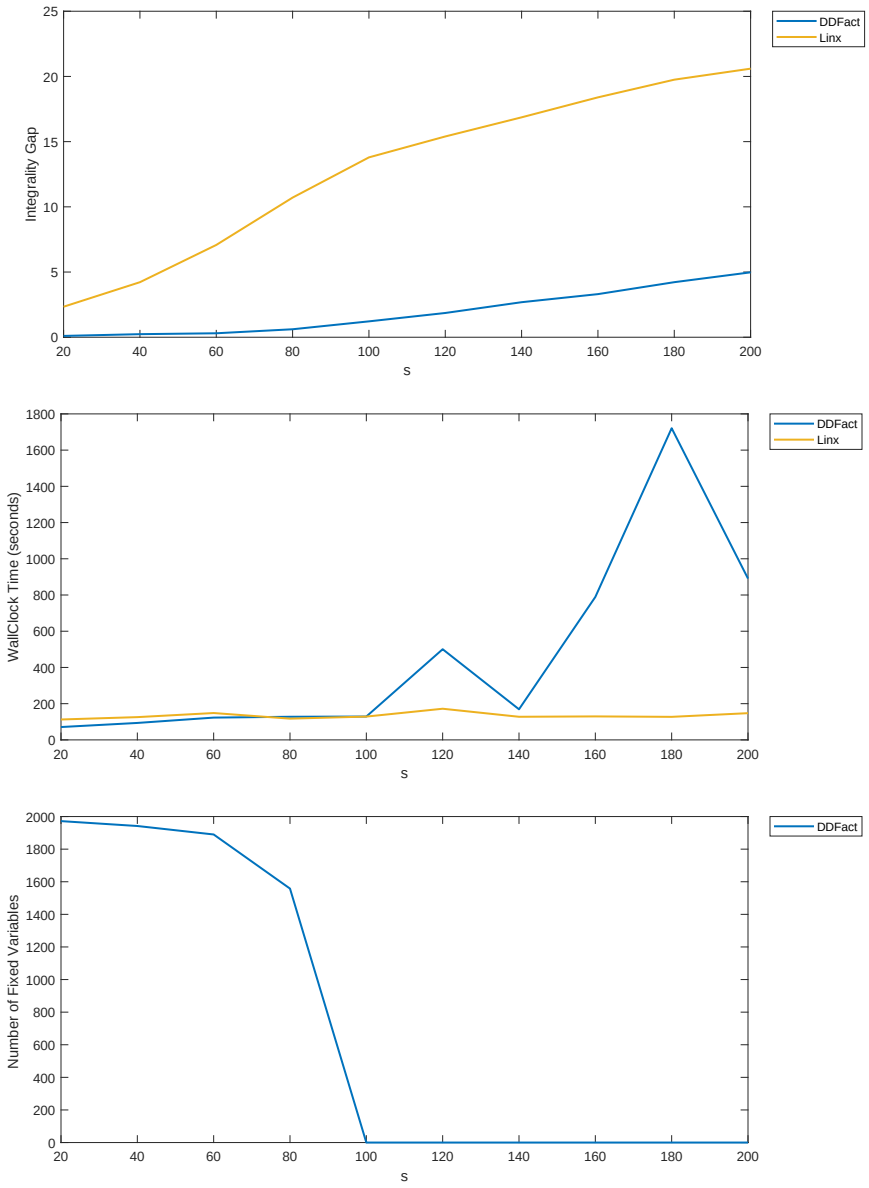

Figure 3.4: Bounds/times comparison and effect of the variable-fixing methodology for $n = 2000$

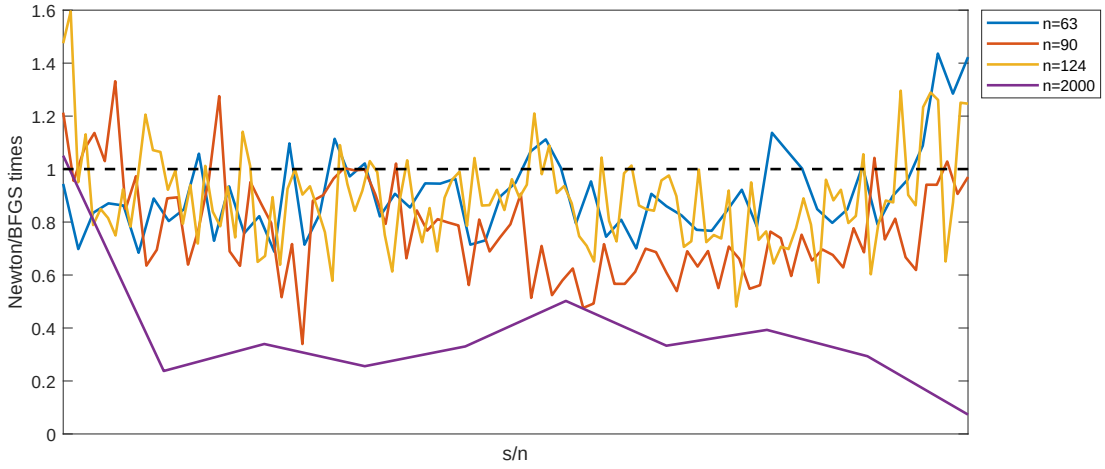

Figure 3.5: Newton/BFGS time for linx

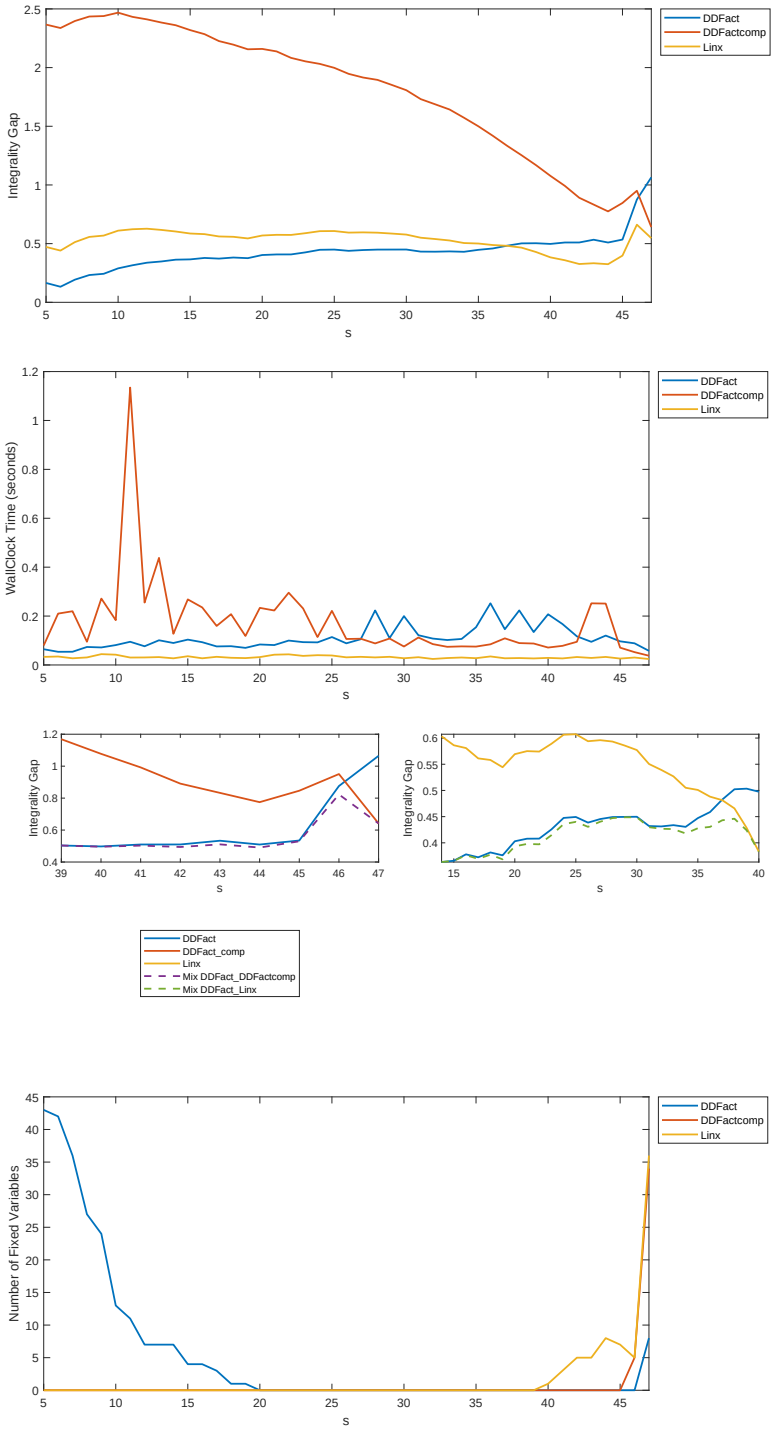

Figure 3.6: Bounds/times comparison and effect of the mixing and variable-fixing methodologies for $n = 63$ with 5 side constraints (CMESP)

CHAPTER 4

Generalized Scaling for the Constrained Maximum-Entropy Sampling Problem

An early/short version of the work in this chapter was published as:

Zhongzhu Chen, Marcia Fampa, Jon Lee. Generalized scaling for the constrained maximum-entropy sampling problem. *Proceedings of ACDA 2023*, 110-118. https://doi.org/10.1137/1.9781611977714.10

A complete version was distributed as:

Zhongzhu Chen, Marcia Fampa, Jon Lee. Generalized scaling for the constrained maximum-entropy sampling problem. https://arxiv.org/abs/2302.04934

4.1 Introduction

A standard and computationally-important bound-enhancement technique for CMESP is scaling, as introduced in chapter 1. Scaling adjusts the shape of continuous relaxations to reduce the gaps between the upper bounds and the optimal value. We extend this technique to *generalized scaling*, employing a positive vector of parameters, which allows much more flexibility and thus significantly reduces the gaps further. Specifically, we generalize the idea of scaling to the vector case and apply it to three different upper bounds: the BQP bound, as well as the state-of-the-art linx and factorization bounds. Throughout, we let $\Upsilon := (\gamma_1, \gamma_2, \ldots, \gamma_n)^\mathsf{T} \in \mathbb{R}^n_{++}$ be a "scaling vector". We refer to our technique as *g-scaling* (i.e., *general scaling*) and the corresponding bounds as *g-scaled* (i.e., *generalized scaled*), and when all elements of Υ are equal, we say *o-scaling* (i.e., *ordinary scaling*) and *o-scaled* (i.e., *ordinary scaled*). If all elements of Υ are equal to 1, we say *un-scaled*. In general, setting all of the elements of Υ to be equal, g-scaling reduces to o-scaling. This means that g-scaling can provide an upper bound that is at least as good as o-scaling. Moreover, as we will see later, g-scaling can often provide significantly improved upper bounds compared to o-scaling. We

give mathematical results aimed at supporting algorithmic methods for computing optimal generalized scalings, and we give computational results demonstrating the performance of generalized scaling on benchmark problem instances.

In §4.2, we introduce the g-scaled BQP bound and establish its convexity in the log of the scaling vector, generalizing an important and practically-useful result for o-scaling (see (Chen, Fampa, Lambert, and Lee, 2021, Theorem 11)). In §4.3, we introduce the g-scaled linx bound and establish its convexity in the log of the scaling vector, generalizing another very important and practically-useful result for o-scaling (see (Chen, Fampa, Lambert, and Lee, 2021, Theorem 18)). These convexity results are key for the tractability of globally optimizing the scaling, something that we do not have for general bound "masking"[1] (see (Anstreicher and Lee, 2004; Burer and Lee, 2007) for this, in the context of the "spectral bound"). In §4.4, we introduce the "g-scaled factorization bound". Here, we also establish "generalized-differentialbility" results for the factorization bound (for the first time), which are essential for the fast and stable calculation of the factorization bound (even using general-purpose nonlinear-optimization software) and for globally optimizing the scaling vector. We are also able to establish that for MESP, the all-ones vector is a stationary point for the "factorization bound" as a function of the scaling vector. Therefore, in contrast to the "BQP bound" and the "linx bound", g-scaling is unlikely to help the factorization bound for MESP. Despite this, through numerical experiments, we observe that g-scaling can significantly improve the "factorization bound" for CMESP, while o-scaling cannot help it (see (Chen, Fampa, and Lee, 2023, Theorem 2.1)). In §4.6, we present results of computational experiments, demonstrating the improvements on upper bounds and on the number of variables that can be fixed (using convex duality) due to g-scaling. In §4.7, we make some brief concluding remarks.

4.2 g-scaled BQP bound

The *g-scaled BQP bound* is defined as follows. For $\Upsilon \in \mathbb{R}^n_{++}$ and $(x, X) \in P_{\text{BQP}}(n, s)$, we now define

$$f_{\text{BQP}}(x, X; \Upsilon) := \text{ldet}\left((\text{Diag}(\Upsilon) C \,\text{Diag}(\Upsilon)) \circ X + \text{Diag}(\mathbf{e} - x) \right) - 2 \textstyle\sum_{i=1}^n x_i \log \gamma_i \,,$$

[1] This is a related bound-improvement technique where we preprocess C by taking its Hadamard product with a correlation matrix.

with domain

$$\operatorname{dom}(f_{\text{BQP}};\Upsilon) := \left\{(x,X)\in\mathbb{R}^n\times\mathbb{S}^n \;:\; \left(\operatorname{Diag}(\Upsilon)C\operatorname{Diag}(\Upsilon)\right)\circ X + \operatorname{Diag}(\mathbf{e}-x)\succ 0\right\}.$$

The *g-scaled BQP bound* is defined as

$$z_{\text{BQP}}(\Upsilon) := \max\left\{f_{\text{BQP}}(x,X;\Upsilon) \;:\; (x,X)\in P_{\text{BQP}}(n,s)\right\}. \tag{gscaling-BQP}$$

We say x is feasible to gscaling-BQP if x satisfies all the constraints in gscaling-BQP.

Note that we can interpret gscaling-BQP as applying the un-scaled BQP bound to the symmetrically-scaled matrix $\operatorname{Diag}(\Upsilon)C\operatorname{Diag}(\Upsilon)$, and then correcting by $-2\sum_{i=1}^n x_i\log\gamma_i$.

Theorem 4.1. *For all $\Upsilon\in\mathbb{R}^n_{++}$, the following hold:*

4.1.i. $z_{BQP}(\Upsilon)$ is a valid upper bound for the optimal value of CMESP, i.e., $z(C,s,A,b)\le z_{BQP}(\Upsilon)$;

4.1.ii. the function $f_{BQP}(x,X;\Upsilon)$ is concave in (x,X) on $\operatorname{dom}(f_{BQP};\Upsilon)$ and continuously differentiable in (x,X,Υ) on $\operatorname{dom}(f_{BQP};\Upsilon)\times\mathbb{R}^n_{++}$;

4.1.iii. for fixed $(x,X)\in\operatorname{dom}(f_{BQP};\Upsilon)$, $f_{BQP}(x,X;\Upsilon)$ is convex in $\log\Upsilon$, and thus $z_{BQP}(\Upsilon)$ is convex in $\log\Upsilon$.

Remark. *(Anstreicher, 2018) established Theorem 4.1.*i *for $\Upsilon:=\gamma\mathbf{e}$, with $\gamma\in\mathbb{R}_{++}$. We generalize this result to $\Upsilon\in\mathbb{R}^n_{++}$. The concavity in Theorem 4.1.*ii *is a result of (Anstreicher, 2018), with details filled in by (Fampa and Lee, 2022, Section 3.6.1). Theorem 4.1.*iii *significantly generalizes a result of (Chen, Fampa, Lambert, and Lee, 2021), where it is established only for o-scaling: i.e., on $\{\Upsilon:=\gamma\mathbf{e} \;:\; \gamma\in\mathbb{R}_{++}\}$. The proof of Theorem 4.1.*iii *requires new techniques (see the proof below). Additionally, the result is quite important as it enables the use of readily available quasi-Newton methods (like BFGS) for finding the globally-optimal g-scaling vector for the gscaling-BQP bound.*

Proof of Theorem 4.1. 4.1.i: It is enough to prove that there is a feasible solution to gscaling-BQP with objective value equal to the optimal value of CMESP. In fact, let $x^*\in\{0,1\}^n$ be an optimal solution to CMESP with support $S(x^*)$, and define $X^*:=x^*(x^*)^{\mathsf{T}}$. Without loss of generality, we assume that $S(x^*)=\{1,\ldots,s\}$, i.e., $x^*=\begin{pmatrix}\mathbf{e}^s\\0\end{pmatrix}$ and $X^*=\begin{pmatrix}I_s & 0\\0 & 0\end{pmatrix}$. Clearly, $(x^*,X^*)\in\operatorname{dom}(f_{\text{BQP}};\Upsilon)$ and is feasible to gscaling-BQP. Let

$\Upsilon_{S(x^*)}$ be the sub-vector of Υ indexed by $S(x^*)$, then

$$\begin{aligned}
&f_{\text{BQP}}(x^*, X^*; \Upsilon) \\
&= \operatorname{ldet}\left(\left(\operatorname{Diag}(\Upsilon) C \operatorname{Diag}(\Upsilon)\right) \circ X^* + \operatorname{Diag}(\mathbf{e} - x^*)\right) - 2\sum_{i=1}^n x_i^* \log \gamma_i \\
&= \operatorname{ldet}\begin{pmatrix} \Upsilon_{S(x^*)} C\left(S(x^*), S(x^*)\right) \Upsilon_{S(x^*)} & 0 \\ 0 & I_{n-s} \end{pmatrix} - 2\sum_{i \in S(x^*)} x_i^* \log \gamma_i \\
&= \operatorname{ldet} C\left(S(x^*), S(x^*)\right).
\end{aligned}$$

4.1.ii: The concavity is essentially a result of (Anstreicher, 2018), with details filled in by (Fampa and Lee, 2022, Section 3.6.1). The continuous differentiability comes from the analyticity of $f_{\text{BQP}}(x, X; \Upsilon)$ in $(x, X, \Upsilon) \in \operatorname{dom}(f_{\text{BQP}}; \Upsilon) \times \mathbb{R}^n_{++}$.

4.1.iii: We sketch the proof first:

1. for fixed $(x, X) \in \operatorname{dom}(f_{\text{BQP}}; \Upsilon)$, we derive the Hessian of $f_{\text{BQP}}(x, X; \Upsilon)$ with respect to $\log \Upsilon$ and show that it is positive-semidefinite, which implies the convexity of $f_{\text{BQP}}(x, X; \Upsilon)$ in $\log \Upsilon$;
2. The convexity of $z_{\text{BQP}}(\Upsilon)$ in $\log \Upsilon$ then follows because $z_{\text{BQP}}(\Upsilon)$ is the pointwise maximum of $f_{\text{BQP}}(x, X; \Upsilon)$ over feasible (x, X) for gscaling-BQP in domain $\operatorname{dom}(f_{\text{BQP}}; \Upsilon)$.

The detailed proof is as follows. For convenience, let

$$\begin{aligned}
F_{\text{BQP}}(x, X; \Upsilon) &:= \left(\operatorname{Diag}(\Upsilon) C \operatorname{Diag}(\Upsilon)\right) \circ X + \operatorname{Diag}(\mathbf{e} - x), \text{ and} \\
A_{\text{BQP}}(X; \Upsilon) &:= \left(\operatorname{Diag}(\Upsilon) C \operatorname{Diag}(\Upsilon)\right) \circ X.
\end{aligned}$$

In the following derivation, we will consider $(x, X) \in \operatorname{dom}\left(f_{\text{BQP}}; \Upsilon\right)$ fixed, and regard Υ as a variable. Thus, for simplicity, we write $F_{\text{BQP}}(x, X; \Upsilon)$ and $A_{\text{BQP}}(X; \Upsilon)$ as $F_{\text{BQP}}(\Upsilon)$ and $A_{\text{BQP}}(\Upsilon)$, respectively.

Let $\check{x} := x - \mathbf{e}$, and we use the identities

$$F_{\text{BQP}}(\Upsilon)^{-1} A_{\text{BQP}}(\Upsilon) = I + F_{\text{BQP}}(\Upsilon)^{-1} \operatorname{Diag}(\check{x}), \tag{4.1}$$

$$A_{\text{BQP}}(\Upsilon) F_{\text{BQP}}(\Upsilon)^{-1} = I + \operatorname{Diag}(\check{x}) F_{\text{BQP}}(\Upsilon)^{-1}. \tag{4.2}$$

We first derive the gradient of $f_{\text{BQP}}(x, X; \Upsilon)$ with respect to Υ.

$$\begin{aligned}
\tfrac{\partial f_{\text{BQP}}(x,X;\Upsilon)}{\partial \gamma_i} &= F_{\text{BQP}}(\Upsilon)^{-1} \bullet \tfrac{\partial A_{\text{BQP}}(\Upsilon)}{\partial \gamma_i} - 2\tfrac{x_i}{\gamma_i} \\
&= F_{\text{BQP}}(\Upsilon)^{-1} \bullet \tfrac{1}{\gamma_i}\left(E_{ii}A_{\text{BQP}}(\Upsilon) + A_{\text{BQP}}(\Upsilon)E_{ii}\right) - 2\tfrac{x_i}{\gamma_i} \\
&= \tfrac{1}{\gamma_i}\left(A_{\text{BQP}}(\Upsilon)_{i\cdot}F_{\text{BQP}}(\Upsilon)_{\cdot i}^{-1} + F_{\text{BQP}}(\Upsilon)_{i\cdot}^{-1}A_{\text{BQP}}(\Upsilon)_{\cdot i} - 2x_i\right) \\
&= \tfrac{1}{\gamma_i}\left(2F_{\text{BQP}}(\Upsilon)_{i\cdot}^{-1}A_{\text{BQP}}(\Upsilon)_{\cdot i} - 2x_i\right),
\end{aligned}$$

where the last identity follows from the symmetry of $F_{\text{BQP}}(\Upsilon)$ and $A_{\text{BQP}}(\Upsilon)$. Then, applying (4.1), we obtain

$$\tfrac{\partial f_{\text{BQP}}(x,X;\Upsilon)}{\partial \Upsilon} = 2\,\text{Diag}(\Upsilon)^{-1}\left(\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) - \check{x}\right).$$

Next, we derive the Hessian of $f_{\text{BQP}}(x, X; \Upsilon)$ with respect to Υ. Note that

$$\begin{aligned}
&\tfrac{1}{2}\tfrac{\partial^2 f_{\text{BQP}}(x,X;\Upsilon)}{\partial \Upsilon \partial \gamma_i} = \tfrac{1}{2}\tfrac{\partial}{\partial \gamma_i}\left(\tfrac{\partial f_{\text{BQP}}(x,X;\Upsilon)}{\partial \Upsilon}\right) \\
&= \tfrac{\partial\,\text{Diag}(\Upsilon)^{-1}}{\partial \gamma_i}\left(\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) - \check{x}\right) \\
&\quad + \text{Diag}(\Upsilon)^{-1}\tfrac{\partial\left(\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) - \check{x}\right)}{\partial \gamma_i} \\
&= \tfrac{\partial\,\text{Diag}(\Upsilon)^{-1}}{\partial \gamma_i}\left(\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) - \check{x}\right) \\
&\quad - \text{Diag}(\Upsilon)^{-1}\,\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\tfrac{\partial F_{\text{BQP}}(\Upsilon)}{\partial \gamma_i}F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) \\
&= -\tfrac{1}{\gamma_i^2}E_{ii}\left(\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) - \check{x}\right) \\
&\quad - \text{Diag}(\Upsilon)^{-1}\,\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\tfrac{\left(E_{ii}A_{\text{BQP}}(\Upsilon) + A_{\text{BQP}}(\Upsilon)E_{ii}\right)}{\gamma_i}F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right).
\end{aligned}$$

The first term in this last expression can be reformulated as

$$\begin{aligned}
&-\tfrac{1}{\gamma_i^2}E_{ii}\left(\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) - \check{x}\right) \\
&= -\,\text{Diag}\left(\Upsilon\right)^{-1}\text{Diag}\left(\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) - \check{x}\right)\text{Diag}\left(\Upsilon\right)^{-1}\mathbf{e}_i\,,
\end{aligned}$$

while for the second term, we use (4.1) and (4.2), and we obtain

$$\begin{aligned}
&\text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\left(E_{ii}A_{\text{BQP}}(\Upsilon) + A_{\text{BQP}}(\Upsilon)E_{ii}\right)F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) \\
&= \text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}E_{ii}\,\text{Diag}(\check{x})\right) + \text{diag}\left(E_{ii}F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) \\
&\quad + \text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}E_{ii}\,\text{Diag}(\check{x})F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right) \\
&\quad + \text{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})E_{ii}F_{\text{BQP}}(\Upsilon)^{-1}\,\text{Diag}(\check{x})\right)
\end{aligned}$$

$$
\begin{aligned}
&= 2\,(x_i - 1)\left(\left(F_{\text{BQP}}(\Upsilon)^{-1}\right)_{ii}\mathbf{e}_i\right) \\
&\quad + 2\,(x_i - 1)\left(\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}E_{ii}F_{\text{BQP}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)\right) \\
&= 2\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\right)\right)\mathbf{e}_i \\
&\quad + 2\operatorname{Diag}(\check{x})\left(F_{\text{BQP}}(\Upsilon)^{-1}\circ F_{\text{BQP}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\mathbf{e}_i\,,
\end{aligned}
$$

which implies that

$$
\begin{aligned}
&-\operatorname{Diag}(\Upsilon)^{-1}\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\tfrac{\left(E_{ii}A_{\text{BQP}}(\Upsilon)+A_{\text{BQP}}(\Upsilon)E_{ii}\right)}{\gamma_i}F_{\text{BQP}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&= -2\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\right)\right)\operatorname{Diag}(\Upsilon)^{-1}\mathbf{e}_i \\
&\quad - 2\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\left(F_{\text{BQP}}(\Upsilon)^{-1}\circ F_{\text{BQP}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\operatorname{Diag}(\Upsilon)^{-1}\mathbf{e}_i\,.
\end{aligned}
$$

Then, we obtain

$$
\begin{aligned}
&\tfrac{\partial^2 f_{\text{BQP}}(x,X;\Upsilon)}{\partial\Upsilon^2} \\
&= -2\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)-\check{x}\right)\operatorname{Diag}(\Upsilon)^{-1} \\
&\quad - 4\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\right)\right)\operatorname{Diag}(\Upsilon)^{-1} \\
&\quad - 4\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\left(F_{\text{BQP}}(\Upsilon)^{-1}\circ F_{\text{BQP}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\operatorname{Diag}(\Upsilon)^{-1}.
\end{aligned}
$$

Finally, we have

$$
\begin{aligned}
&\tfrac{\partial^2 f_{\text{BQP}}(x,X;\Upsilon)}{\partial(\log\Upsilon)^2} = \operatorname{Diag}(\Upsilon)\tfrac{\partial f_{\text{BQP}}(x,X;\Upsilon)}{\partial\Upsilon} + \operatorname{Diag}(\Upsilon)\tfrac{\partial^2 f_{\text{BQP}}(x,X;\Upsilon)}{\partial\Upsilon^2}\operatorname{Diag}(\Upsilon) \\
&= -4\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\right)\right) \\
&\quad - 4\operatorname{Diag}(\check{x})\left(F_{\text{BQP}}(\Upsilon)^{-1}\circ F_{\text{BQP}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x}) \\
&= 4\operatorname{Diag}(\mathbf{e}-x)\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{BQP}}(\Upsilon)^{-1}\right)\right) \\
&\quad - 4\operatorname{Diag}(\mathbf{e}-x)\left(F_{\text{BQP}}(\Upsilon)^{-1}\circ F_{\text{BQP}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\mathbf{e}-x).
\end{aligned}
$$

Next, we will show the positive semidefiniteness of $\frac{\partial^2 f_{\text{BQP}}(x,X;\Upsilon)}{\partial(\log\Upsilon)^2}$ for all $0 \le x \le \mathbf{e}, X \succeq 0$ such that $(x,X) \in \operatorname{dom}(f_{\text{BQP}};\Upsilon)$. Note that we will not require (x,X) to be feasible for gscaling-BQP. We analyse two cases.

Case 1: when $0 \le x < \mathbf{e}$ and $X \succeq 0$, let $D_{\text{BQP}}(x) := (\operatorname{Diag}(\mathbf{e}-x))^{1/2} \succ 0$, and let $H_{\text{BQP}}(x,X;\Upsilon) := (D_{\text{BQP}}(x))^{-1}A_{\text{BQP}}(\Upsilon)(D_{\text{BQP}}(x))^{-1} \succeq 0$. Again, for simplicity, we write $D_{\text{BQP}}(x)$ and $H_{\text{BQP}}(x,X;\Upsilon)$ as D_{BQP} and $H_{\text{BQP}}(\Upsilon)$, respectively. First, we note

that

$$D_{\text{BQP}}F_{\text{BQP}}(\Upsilon)^{-1}D_{\text{BQP}} = \left(D_{\text{BQP}}^{-1}A_{\text{BQP}}(\Upsilon)D_{\text{BQP}}^{-1} + I\right)^{-1}.$$

Then, we have

$$\begin{aligned}
&\tfrac{1}{4}\tfrac{\partial^2 f_{\text{BQP}}(x,X;\Upsilon)}{\partial(\log\Upsilon)^2} \\
&= \text{Diag}\left(\text{diag}\left(D_{\text{BQP}}F_{\text{BQP}}(\Upsilon)^{-1}D_{\text{BQP}}\right)\right) - \\
&\quad \text{Diag}\left(D_{\text{BQP}}F_{\text{BQP}}(\Upsilon)^{-1}D_{\text{BQP}}\right) \circ \text{Diag}\left(D_{\text{BQP}}F_{\text{BQP}}(\Upsilon)^{-1}D_{\text{BQP}}\right) \\
&= \left(D_{\text{BQP}}F_{\text{BQP}}(\Upsilon)^{-1}D_{\text{BQP}}\right) \circ I - \\
&\quad \text{Diag}\left(D_{\text{BQP}}F_{\text{BQP}}(\Upsilon)^{-1}D_{\text{BQP}}\right) \circ \text{Diag}\left(D_{\text{BQP}}F_{\text{BQP}}(\Upsilon)^{-1}D_{\text{BQP}}\right) \\
&= \left(H_{\text{BQP}}(\Upsilon) + I\right)^{-1} \circ I - \left(H_{\text{BQP}}(\Upsilon) + I\right)^{-1} \circ \left(H_{\text{BQP}}(\Upsilon) + I\right)^{-1} \\
&= \left(H_{\text{BQP}}(\Upsilon) + I\right)^{-1} \circ \left(I - \left(H_{\text{BQP}}(\Upsilon) + I\right)^{-1}\right) \succeq 0.
\end{aligned}$$

The last inequality holds because $H_{\text{BQP}}(\Upsilon) + I \succ 0$ and the Schur Product Theorem.

Case 2: now, we discuss the general case $0 \leq x \leq \mathbf{e}, X \succeq 0$. Note that for $\Upsilon \in \mathbb{R}^n_{++}$, $\frac{\partial f^2_{\text{BQP}}(x,X;\Upsilon)}{\partial(\log\Upsilon)^2}$ is analytic in $0 \leq x \leq \mathbf{e}, X \succeq 0$ such that $(x,X) \in \text{dom}\,(f_{\text{BQP}};\Upsilon)$. Therefore, given $0 \leq x \leq \mathbf{e}, X \succeq 0$, assume that $\frac{\partial f^2_{\text{BQP}}(x,X;\Upsilon)}{\partial(\log\Upsilon)^2} \not\succeq 0$. Then by the analyticity (continuity) of $\frac{\partial f^2_{\text{BQP}}(x,X;\Upsilon)}{\partial(\log\Upsilon)^2}$, there exists small enough $\epsilon > 0$ such that for any $0 \leq x' \leq \mathbf{e}, X' \succeq 0$ in the intersection of the neighbourhood

$$\mathcal{N}_\epsilon(x,X) := \left\{(x',X') : \|x - x'\|_\infty + \|X - X'\|_F \leq \epsilon\right\},$$

(where $\|\cdot\|_\infty$ is the vector infinity-norm, and $\|\cdot\|_F$ is the Frobenius norm) and $\{(x',X') : 0 \leq x' \leq \mathbf{e}, X' \succeq 0, (x',X') \in \text{dom}\,(f_{\text{BQP}};\Upsilon)\}$, we have $\frac{\partial f^2_{\text{BQP}}(x',X';\Upsilon)}{\partial(\log\Upsilon)^2} \not\succeq 0$. On the other hand, this intersection contains some (x',X') such that $0 \leq x' < \mathbf{e}$, $X' \succeq 0$, e.g. $(x',X') = (x - \sum_{i:x_i=1} \epsilon e_i, X)$. This is a contradiction to Case 1.

In conclusion, for each fixed $(x,X) \in \{(x,X) : 0 \leq x \leq \mathbf{e}, X \succeq 0, (x,X) \in \text{dom}\,(f_{\text{BQP}};\Upsilon)\}$, we have that $f_{\text{BQP}}(x,X;\Upsilon)$ is convex in $\log\Upsilon$. In particular, for $(x,X) \in \text{dom}\,(f_{\text{BQP}};\Upsilon)$ and feasible to gscaling-BQP, $f_{\text{BQP}}(x,X;\Upsilon)$ is convex in $\log\Upsilon$. Finally, as $z_{\text{BQP}}(\Upsilon)$ is the point-wise maximum of $f_{\text{BQP}}(x,X;\Upsilon)$ over all such (x,X), then $z_{\text{BQP}}(\Upsilon)$ is convex in $\log\Upsilon$.

□

4.3 g-scaled linx bound

The *g-scaled linx bound* is defined as follows. For $\Upsilon \in \mathbb{R}^n_{++}$ and $x \in [0,1]^n$, we now define

$$f_{\text{linx}}(x;\Upsilon) := \tfrac{1}{2}\left(\operatorname{ldet}\left(\operatorname{Diag}(\Upsilon)C\operatorname{Diag}(x)C\operatorname{Diag}(\Upsilon) + \operatorname{Diag}(\mathbf{e}-x)\right)\right) - \textstyle\sum_{i=1}^n x_i \log\gamma_i$$

with

$$\operatorname{dom}(f_{\text{linx}};\Upsilon) := \left\{x \in \mathbb{R}^n : \operatorname{Diag}(\Upsilon)C\operatorname{Diag}(x)C\operatorname{Diag}(\Upsilon) + \operatorname{Diag}(\mathbf{e}-x) \succ 0\right\}.$$

We then define the *g-scaled linx bound*

$$z_{\text{linx}}(\Upsilon) := \max\left\{f_{\text{linx}}(x;\Upsilon) \;:\; x \in P_{\text{linx}}(n,s)\right\}. \tag{gscaling-linx}$$

We say that x is feasible to gscaling-linx if x satisfies all the constraints in gscaling-linx.

It is very important to note, in contrast to g-scaling for the gscaling-BQP bound, that we are *not* applying the ordinary gscaling-linx bound to a symmetric scaling of C. In this way, g-scaling for the gscaling-linx bound is more subtle. Rather, we are symmetrically scaling $\operatorname{Diag}(\Upsilon)C\operatorname{Diag}(x)C\operatorname{Diag}(\Upsilon)$. This point would not apply to o-scaling, as scalars commute through matrix multiplication.

Theorem 4.2. *For all $\Upsilon \in \mathbb{R}^n_{++}$ in gscaling-linx, the following hold:*

4.2.i. *$z_{linx}(\Upsilon)$ is a valid upper bound for the optimal value of CMESP, i.e., $z(C,s,A,b) \le z_{linx}(\Upsilon)$;*

4.2.ii. *the function $f_{linx}(x;\Upsilon)$ is concave in x on $\operatorname{dom}(f_{linx};\Upsilon)$ and continuously differentiable in (x,Υ) on $\operatorname{dom}(f_{linx};\Upsilon) \times \mathbb{R}^n_{++}$;*

4.2.iii. *for fixed $x \in \operatorname{dom}(f_{linx};\Upsilon)$, $f_{linx}(x;\Upsilon)$ is convex in $\log\Upsilon$, and thus $z_{linx}(\Upsilon)$ is convex in $\log\Upsilon$.*

Remark. *(Anstreicher, 2020) established Theorem 4.2.*i *for $\Upsilon := \gamma\mathbf{e}$, with $\gamma \in \mathbb{R}_{++}$. We generalize this result to $\Upsilon \in \mathbb{R}^n_{++}$. The concavity in Theorem 4.2.*ii *is a result of (Anstreicher, 2020), with details filled in by (Fampa and Lee, 2022). Theorem 4.2.*iii *generalizes a result of (Chen, Fampa, Lambert, and Lee, 2021), where it is established only for o-scaling: i.e., on $\{\Upsilon = \gamma\mathbf{e} \;:\; \gamma \in \mathbb{R}_{++}\}$. The proof of Theorem 4.2.*iii *requires new techniques (see the below). Additionally, the result is quite important as it enables the use of readily available quasi-Newton methods (like BFGS) for finding the globally optimal g-scaling for the gscaling-linx*

bound.

Proof of Theorem 4.2. 4.2.i: It is enough to prove that there is a feasible solution to gscaling-linx with objective value equal to the optimal value of CMESP. In fact, let $x^* \in \{0,1\}^n$ be one optimal solution to CMESP with support $S(x^*)$, and define $X^* := x^*(x^*)^\mathsf{T}$. Without loss of generality, we assume that $S(x^*) = \{1, \ldots, s\}$, i.e., $x^* = \begin{pmatrix} \mathbf{e}_s \\ 0 \end{pmatrix}$. Let $T(x^*) := N \backslash S(x^*)$ be the complementary set of $S(x^*)$. For convenience, we denote $\tilde{C} := \mathrm{Diag}(\Upsilon)C$, $\tilde{C}_{ST} := \tilde{C}(S(x^*), T(x^*))$, and so on. Note that $\tilde{C}$ is not symmetric. Also, note that $\tilde{C}$ depends on Υ, and $\tilde{C}_{ST}$ depends on $\Upsilon, x^*, S(x^*)$, and $T(x^*)$. We can write

$$\tilde{C}\,\mathrm{Diag}(x)\tilde{C}^\mathsf{T} = \begin{pmatrix} \tilde{C}_{SS} & \tilde{C}_{ST} \\ \tilde{C}_{TS} & \tilde{C}_{TT} \end{pmatrix} \begin{pmatrix} I_s & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{C}_{SS}^\mathsf{T} & \tilde{C}_{TS}^\mathsf{T} \\ \tilde{C}_{ST}^\mathsf{T} & \tilde{C}_{TT}^\mathsf{T} \end{pmatrix} = \begin{pmatrix} \tilde{C}_{SS}\tilde{C}_{SS}^\mathsf{T} & \tilde{C}_{SS}\tilde{C}_{TS}^\mathsf{T} \\ \tilde{C}_{TS}^T\tilde{C}_{SS}^\mathsf{T} & \tilde{C}_{TS}\tilde{C}_{TS}^\mathsf{T} \end{pmatrix},$$

and therefore

$$\tilde{C}\,\mathrm{Diag}(x)\tilde{C}^\mathsf{T} + \mathrm{Diag}(\mathbf{e} - x) = \begin{pmatrix} \tilde{C}_{SS}\tilde{C}_{SS}^\mathsf{T} & \tilde{C}_{SS}\tilde{C}_{TS}^\mathsf{T} \\ \tilde{C}_{TS}^T\tilde{C}_{SS}^\mathsf{T} & \tilde{C}_{TS}\tilde{C}_{TS}^\mathsf{T} + I_{n-s} \end{pmatrix}.$$

Applying the well-known Schur-complement determinant formula, we then obtain

$$\begin{aligned} &\mathrm{ldet}\left(\tilde{C}\,\mathrm{Diag}(x)\tilde{C}^\mathsf{T} + \mathrm{Diag}(\mathbf{e} - x)\right) \\ &= 2\,\mathrm{ldet}\,\tilde{C}_{SS} + \mathrm{ldet}\left(\tilde{C}_{TS}^\mathsf{T}\tilde{C}_{TS} + I_{n-s} - \tilde{C}_{TS}^\mathsf{T}\tilde{C}_{SS}^\mathsf{T}\tilde{C}_{SS}^{-\mathsf{T}}\tilde{C}_{SS}^{-1}\tilde{C}_{SS}\tilde{C}_{TS}^\mathsf{T}\right) \\ &= 2\,\mathrm{ldet}\,\tilde{C}_{SS}\,. \end{aligned}$$

Let $\Upsilon_{S(x^*)}$ be the sub-vector of Υ indexed by $S(x^*)$. Then, we have

$$\begin{aligned} f_{\text{linx}}(x^*;\Upsilon) &= \tfrac{1}{2}\,\mathrm{ldet}\left(\tilde{C}\,\mathrm{Diag}(x)\tilde{C}^\mathsf{T} + \mathrm{Diag}(\mathbf{e} - x)\right) - \textstyle\sum_{i\in N} x_i^* \log\gamma_i \\ &= \mathrm{ldet}\,\tilde{C}_{SS} - \textstyle\sum_{i\in S(x^*)} \log\gamma_i \\ &= \mathrm{ldet}\left(\,\mathrm{Diag}\left(\Upsilon_{S(x^*)}\right) C\left(S(x^*), S(x^*)\right)\right) - \textstyle\sum_{i\in S(x^*)} \log\gamma_i \\ &= \mathrm{ldet}\,C\left(S(x^*), S(x^*)\right). \end{aligned}$$

4.2.ii: The concavity is essentially a result of (Anstreicher, 2020), with details filled in by (Fampa and Lee, 2022, Section 3.3.1). The continuous differentiability comes from the analyticity of $f_{\text{linx}}(x;\Upsilon)$ in $(x,\Upsilon) \in \mathrm{dom}\,(f_{\text{linx}};\Upsilon) \times \mathbb{R}^n_{++}$.

4.2.iii: We sketch the proof first:

1. for fixed $x \in \mathrm{dom}\,(f_{\mathrm{linx}}; \Upsilon)$, we derive the Hessian of $f_{\mathrm{linx}}(x; \Upsilon)$ with respect to $\log \Upsilon$ and show that it is positive-semidefinite, which implies the convexity of $f_{\mathrm{linx}}(x; \Upsilon)$ in $\log \Upsilon$;
2. The convexity of $z_{\mathrm{linx}}(\Upsilon)$ in $\log \Upsilon$ then follows because $z_{\mathrm{linx}}(\Upsilon)$ is the point-wise maximum of $f_{\mathrm{linx}}(x; \Upsilon)$ over feasible x for gscaling-linx in domain $\mathrm{dom}\,(f_{\mathrm{linx}}; \Upsilon)$.

The detailed proof is as follows: for convenience, let

$$F_{\mathrm{linx}}(x; \Upsilon) := \mathrm{Diag}(\Upsilon) C \,\mathrm{Diag}(x) C \,\mathrm{Diag}(\Upsilon) + \mathrm{Diag}(\mathbf{e} - x), \text{ and}$$
$$A_{\mathrm{linx}}(x; \Upsilon) := \mathrm{Diag}(\Upsilon) C \,\mathrm{Diag}(x) C \,\mathrm{Diag}(\Upsilon).$$

In the following derivation, we will fix x and regard Υ as a variable. Thus, for simplicity, we will write $F_{\mathrm{linx}}(x; \Upsilon)$ and $A_{\mathrm{linx}}(x; \Upsilon)$ as $F_{\mathrm{linx}}(\Upsilon)$ and $A_{\mathrm{linx}}(\Upsilon)$, respectively. Let $\check{x} := x - \mathbf{e}$, and we note that

$$F_{\mathrm{linx}}(\Upsilon)^{-1} A_{\mathrm{linx}}(\Upsilon) = I + F_{\mathrm{linx}}(\Upsilon)^{-1} \,\mathrm{Diag}(\check{x}), \tag{4.3}$$
$$A_{\mathrm{linx}}(\Upsilon) F_{\mathrm{linx}}(\Upsilon)^{-1} = I + \mathrm{Diag}(\check{x}) F_{\mathrm{linx}}(\Upsilon)^{-1}. \tag{4.4}$$

Given $x \in \mathrm{dom}\,(f_{\mathrm{linx}}; \Upsilon)$, we first derive the gradient of $f_{\mathrm{linx}}(x; \Upsilon)$ with respect to Υ. We have

$$\begin{aligned}
\tfrac{\partial f_{\mathrm{linx}}(x;\Upsilon)}{\partial \gamma_i} &= \tfrac{1}{2} F_{\mathrm{linx}}(\Upsilon)^{-1} \bullet \tfrac{\partial A_{\mathrm{linx}}(\Upsilon)}{\partial \gamma_i} - \tfrac{x_i}{\gamma_i} \\
&= \tfrac{1}{2} F_{\mathrm{linx}}(\Upsilon)^{-1} \bullet \tfrac{1}{\gamma_i} \left(E_{ii} A_{\mathrm{linx}}(\Upsilon) + A_{\mathrm{linx}}(\Upsilon) E_{ii} \right) - \tfrac{x_i}{\gamma_i} \\
&= \tfrac{1}{2\gamma_i} \left(\left(A_{\mathrm{linx}}(\Upsilon)_{i\cdot} F_{\mathrm{linx}}(\Upsilon)^{-1}_{\cdot i} + F_{\mathrm{linx}}(\Upsilon)^{-1}_{i\cdot} A_{\mathrm{linx}}(\Upsilon)_{\cdot i} \right) - 2x_i \right) \\
&= \tfrac{1}{\gamma_i} \left(F_{\mathrm{linx}}(\Upsilon)^{-1}_{i\cdot} A_{\mathrm{linx}}(\Upsilon)_{\cdot i} - x_i \right),
\end{aligned}$$

where the last identity follows from the symmetry of $F_{\mathrm{linx}}(\Upsilon)$ and $A_{\mathrm{linx}}(\Upsilon)$. Then, applying (4.3), we obtain

$$\tfrac{\partial f_{\mathrm{linx}}(x;\Upsilon)}{\partial \Upsilon} = \mathrm{Diag}(\Upsilon)^{-1} \left(\mathrm{diag}\left(F_{\mathrm{linx}}(\Upsilon)^{-1} \,\mathrm{Diag}(\check{x}) \right) - \check{x} \right).$$

Next, we derive the Hessian of $f_{\mathrm{linx}}(x; \Upsilon)$ with respect to Υ. We have

$$\begin{aligned}
\tfrac{\partial^2 f_{\mathrm{linx}}(x;\Upsilon)}{\partial \Upsilon \partial \gamma_i} &= \tfrac{\partial}{\partial \gamma_i} \left(\tfrac{\partial f_{\mathrm{linx}}(x;\Upsilon)}{\partial \Upsilon} \right) \\
&= \tfrac{\partial \,\mathrm{Diag}(\Upsilon)^{-1}}{\partial \gamma_i} \left(\mathrm{diag}\left(F_{\mathrm{linx}}(\Upsilon)^{-1} \,\mathrm{Diag}(\check{x}) \right) - \check{x} \right)
\end{aligned}$$

$$
\begin{aligned}
&+ \operatorname{Diag}(\Upsilon)^{-1} \tfrac{\partial\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)-\check{x}\right)}{\partial\gamma_i} \\
&= \tfrac{\partial \operatorname{Diag}(\Upsilon)^{-1}}{\partial\gamma_i}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)-\check{x}\right) \\
&\quad - \operatorname{Diag}(\Upsilon)^{-1}\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\tfrac{\partial F_{\text{linx}}(\Upsilon)}{\partial\gamma_i}F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&= -\tfrac{1}{\gamma_i^2}E_{ii}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)-\check{x}\right) \\
&\quad - \operatorname{Diag}(\Upsilon)^{-1}\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\tfrac{(E_{ii}A_{\text{linx}}(\Upsilon)+A_{\text{linx}}(\Upsilon)E_{ii})}{\gamma_i}F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right).
\end{aligned}
$$

For the first term, we can reformulate

$$
\begin{aligned}
&-\tfrac{1}{\gamma_i^2}E_{ii}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)-\check{x}\right) \\
&= -\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)-\check{x}\right)\operatorname{Diag}(\Upsilon)^{-1}\mathbf{e}_i\,,
\end{aligned}
$$

while for the second term, we can reformulate

$$
\begin{aligned}
&\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\left(E_{ii}A_{\text{linx}}(\Upsilon)+A_{\text{linx}}(\Upsilon)E_{ii}\right)F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&= \operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}E_{ii}\left(I+\operatorname{Diag}(\check{x})F_{\text{linx}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\right) \\
&\quad + \operatorname{diag}\left(\left(I+F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)E_{ii}F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&= \operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}E_{ii}\operatorname{Diag}(\check{x})\right)+\operatorname{diag}\left(E_{ii}F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&\quad + \operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}E_{ii}\operatorname{Diag}(\check{x})F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&\quad + \operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})E_{ii}F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&= 2\left(x_i-1\right)\left(\left(F_{\text{linx}}(\Upsilon)^{-1}\right)_{ii}\mathbf{e}_i\right) \\
&\quad + 2\left(x_i-1\right)\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}E_{ii}F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)\right) \\
&= 2\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\right)\right)\mathbf{e}_i \\
&\quad + 2\operatorname{Diag}(\check{x})\left(F_{\text{linx}}(\Upsilon)^{-1}\circ F_{\text{linx}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\mathbf{e}_i\,,
\end{aligned}
$$

which implies that

$$
\begin{aligned}
&\tfrac{1}{\gamma_i}\operatorname{Diag}(\Upsilon)^{-1}\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\left(E_{ii}A_{\text{linx}}(\Upsilon)+A_{\text{linx}}(\Upsilon)E_{ii}\right)F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right) \\
&= 2\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\right)\right)\operatorname{Diag}(\Upsilon)^{-1}\mathbf{e}_i \\
&\quad + 2\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\left(F_{\text{linx}}(\Upsilon)^{-1}\circ F_{\text{linx}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\operatorname{Diag}(\Upsilon)^{-1}\mathbf{e}_i\,.
\end{aligned}
$$

Finally, we obtain

$$
\tfrac{\partial^2 f_{\text{linx}}(x;\Upsilon)}{\partial\Upsilon^2} = -\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\right)-\check{x}\right)\operatorname{Diag}(\Upsilon)^{-1}
$$

$$- 2\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\right)\right)\operatorname{Diag}(\Upsilon)^{-1}$$
$$- 2\operatorname{Diag}(\Upsilon)^{-1}\operatorname{Diag}(\check{x})\left(F_{\text{linx}}(\Upsilon)^{-1}\circ F_{\text{linx}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\operatorname{Diag}(\Upsilon)^{-1}.$$

Then, we have

$$\begin{aligned}
\tfrac{\partial^2 f_{\text{linx}}(x;\Upsilon)}{\partial(\log\Upsilon)^2} &= \operatorname{Diag}(\Upsilon)\tfrac{\partial f_{\text{linx}}(x;\Upsilon)}{\partial\Upsilon} + \operatorname{Diag}(\Upsilon)\tfrac{\partial^2 f_{\text{linx}}(x;\Upsilon)}{\partial\Upsilon^2}\operatorname{Diag}(\Upsilon)\\
&= -2\operatorname{Diag}(\check{x})\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\right)\right)\\
&\quad - 2\operatorname{Diag}(\check{x})\left(F_{\text{linx}}(\Upsilon)^{-1}\circ F_{\text{linx}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\check{x})\\
&= 2\operatorname{Diag}(\mathbf{e}-x)\operatorname{Diag}\left(\operatorname{diag}\left(F_{\text{linx}}(\Upsilon)^{-1}\right)\right)\\
&\quad - 2\operatorname{Diag}(\mathbf{e}-x)\left(F_{\text{linx}}(\Upsilon)^{-1}\circ F_{\text{linx}}(\Upsilon)^{-1}\right)\operatorname{Diag}(\mathbf{e}-x).
\end{aligned}$$

Next, we are going to show the positive semidefiniteness of $\frac{\partial^2 f_{\text{linx}}(x;\Upsilon)}{\partial(\log\Upsilon)^2}$ for all $0 \le x \le \mathbf{e}$ such that $x \in \operatorname{dom}(f_{\text{linx}};\Upsilon)$. Note that we will not require x to be feasible to gscaling-linx. We divide the discussion into two cases.

Case 1: when $0 \le x < \mathbf{e}$, let $D_{\text{linx}}(x) := (\operatorname{Diag}(\mathbf{e}-x))^{1/2} \succ 0$, and $H_{\text{linx}}(x;\Upsilon) := (D_{\text{linx}}(x))^{-1} A_{\text{linx}}(\Upsilon)(D_{\text{linx}}(x))^{-1} \succeq 0$. Again for simplicity, we write $D_{\text{linx}}(x)$ and $H_{\text{linx}}(x;\Upsilon)$ as D_{linx} and $H_{\text{linx}}(\Upsilon)$. First, we note

$$D_{\text{linx}}F_{\text{linx}}(\Upsilon)^{-1}D_{\text{linx}} = \left(D_{\text{linx}}^{-1}A_{\text{linx}}(\Upsilon)D_{\text{linx}}^{-1} + I\right)^{-1}.$$

Then, we have

$$\begin{aligned}
&\tfrac{1}{2}\tfrac{\partial^2 f_{\text{linx}}(x;\Upsilon)}{\partial(\log\Upsilon)^2}\\
&= \operatorname{Diag}\left(\operatorname{diag}\left(D_{\text{linx}}F_{\text{linx}}(\Upsilon)^{-1}D_{\text{linx}}\right)\right)\\
&\quad - \operatorname{Diag}\left(D_{\text{linx}}F_{\text{linx}}(\Upsilon)^{-1}D_{\text{linx}}\right)\circ\operatorname{Diag}\left(D_{\text{linx}}F_{\text{linx}}(\Upsilon)^{-1}D_{\text{linx}}\right)\\
&= \left(D_{\text{linx}}F_{\text{linx}}(\Upsilon)^{-1}D_{\text{linx}}\right)\circ I\\
&\quad - \operatorname{Diag}\left(D_{\text{linx}}F_{\text{linx}}(\Upsilon)^{-1}D_{\text{linx}}\right)\circ\operatorname{Diag}\left(D_{\text{linx}}F_{\text{linx}}(\Upsilon)^{-1}D_{\text{linx}}\right)\\
&= (H_{\text{linx}}(\Upsilon)+I)^{-1}\circ I - (H_{\text{linx}}(\Upsilon)+I)^{-1}\circ(H_{\text{linx}}(\Upsilon)+I)^{-1}\\
&= (H_{\text{linx}}(\Upsilon)+I)^{-1}\circ\left(I - (H_{\text{linx}}(\Upsilon)+I)^{-1}\right) \succeq 0.
\end{aligned}$$

The last inequality holds because $H_{\text{BQP}}(\Upsilon) + I \succ 0$ and the Schur Product Theorem.

Case 2: We now discuss general $0 \le x \le \mathbf{e}$. Note that given $\Upsilon \in \mathbb{R}^n_{++}$, $\frac{\partial f^2_{\text{linx}}(x;\Upsilon)}{\partial(\log\Upsilon)^2}$ is analytical in $0 \le x \le \mathbf{e}$ such that $x \in \operatorname{dom}(f_{\text{linx}};\Upsilon)$. Therefore, given $0 \le x \le \mathbf{e}$, assume that $\frac{\partial f^2_{\text{linx}}(x;\Upsilon)}{\partial(\log\Upsilon)^2} \not\succeq 0$. Then by the analyticity (continuity) of $\frac{\partial f^2_{\text{linx}}(x;\Upsilon)}{\partial(\log\Upsilon)^2}$, there

exists small enough $\epsilon > 0$ such that for any $0 \leq x' \leq \mathbf{e}$ in the intersection of neighbourhood $\mathcal{N}_\epsilon(x) := \{x' : \|x - x'\|_\infty \leq \epsilon\}$ (where $\|\cdot\|_\infty$ is the vector infinity norm) and $\{x' : 0 \leq x' \leq \mathbf{e}, x' \in \operatorname{dom}(f_{\text{linx}}; \Upsilon)\}$, we have $\frac{\partial f^2_{\text{linx}}(x;\Upsilon)}{\partial(\log \Upsilon)^2} \not\succeq 0$. On the other hand, this intersection contains some x' such that $0 \leq x' < \mathbf{e}$, e.g. $x' = x - \sum_{i:x_i=1} \epsilon e_i$. This contradicts Case 1.

In conclusion, for each fixed $x \in \{(x, X) : 0 \leq x \leq \mathbf{e}, x \in \operatorname{dom}(f_{\text{linx}}; \Upsilon)\}$, $f_{\text{linx}}(x; \Upsilon)$ is convex in $\log \Upsilon$. In particular, for $x \in \operatorname{dom}(f_{\text{linx}}; \Upsilon)$ and feasible to gscaling-linx, $f_{\text{linx}}(x; \Upsilon)$ is convex in $\log \Upsilon$. Finally, as $z_{\text{linx}}(\Upsilon)$ is the point-wise maximum of $f_{\text{linx}}(x; \Upsilon)$ over all such x, we have that $z_{\text{linx}}(\Upsilon)$ is convex in $\log \Upsilon$.

□

4.4 g-scaled factorization bound

The factorization bound was first analyzed in (Nikolov, 2015), and then developed further in (Li and Xie, 2023) and in (Chen, Fampa, and Lee, 2023) (see (Fampa and Lee, 2022, Section 3.4) for more details). The definition of the factorization bound is based on the following key lemma.

Lemma 4.3. *(see (Nikolov, 2015, Lemma 14)) Let $\lambda \in \mathbb{R}^k_+$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$, and let $0 < s \leq k$. There exists a unique integer ι, with $0 \leq \iota < s$, such that $\lambda_\iota > \frac{1}{s-\iota} \sum_{\ell=\iota+1}^k \lambda_\ell \geq \lambda_{\iota+1}$, with the convention $\lambda_0 := +\infty$.*

Now, suppose that $\lambda \in \mathbb{R}^k_+$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$. Given an integer s with $0 < s \leq k$, let ι be the unique integer defined by Lemma 4.3. We define

$$\phi_s(\lambda) := \sum_{\ell=1}^{\iota} \log \lambda_\ell + (s - \iota) \log \left(\tfrac{1}{s-\iota} \sum_{\ell=\iota+1}^k \lambda_\ell \right).$$

Next, for $X \in \mathbb{S}^k_+$, we define $\Gamma_s(X) := \phi_s(\lambda_1(X), \ldots, \lambda_k(X))$ where $\lambda_1(X) \geq \lambda_2(X) \geq \cdots \geq \lambda_k(X)$ are the eigenvalues of X.

Suppose that the rank of C is $r \geq s$. Then we factorize $C = FF^\mathsf{T}$, with $F \in \mathbb{R}^{n \times k}$, for some k satisfying $r \leq k \leq n$. It has been established (Chen, Fampa, and Lee, 2023, Theorem 2.2) that the value of the factorization bound is independent of the choice of F. Consequently, for the sake of simplicity, while certain terms may feature F in their defining equations, it will not be included as a parameter for such terms.

Now, for $\Upsilon \in \mathbb{R}^n_{++}$ and $x \in [0, 1]^n$, we define

$$F_{\text{DDFact}}(x; \Upsilon) := \sum_{i=1}^n \gamma_i x_i F^\mathsf{T}_{i\cdot} F_{i\cdot}, \text{ and}$$

$$f_{\text{DDFact}}(x;\Upsilon) := \Gamma_s(F_{\text{DDFact}}(x;\Upsilon)) - \sum_{i=1}^n x_i \log \gamma_i \,.$$

Finally, we define the *g-scaled factorization bound*

$$z_{\text{DDFact}}(\Upsilon) := \max\left\{ f_{\text{DDFact}}(x;\Upsilon) \,:\, \mathbf{e}^\mathsf{T} x = s,\ 0 \le x \le \mathbf{e},\ Ax \le b \right\}. \qquad \text{(gscaling-DDFact)}$$

The reason for the nomenclature gscaling-DDFact is because it is obtained from the Lagrangian dual of the Lagrangian dual of a nonconvex continuous relaxation of CMESP (see (Chen, Fampa, and Lee, 2023)). Note that

$$F_{\text{DDFact}}(x;\Upsilon) = F^\mathsf{T} \operatorname{Diag}(\sqrt{\Upsilon}) \operatorname{Diag}(x) \operatorname{Diag}(\sqrt{\Upsilon}) F.$$

So, we can interpret gscaling-DDFact as applying the un-scaled gscaling-DDFact bound to the symmetrically-scaled matrix $\operatorname{Diag}(\sqrt{\Upsilon})F \operatorname{Diag}\left(\sqrt{\Upsilon}\right) F^\mathsf{T} \operatorname{Diag}(\sqrt{\Upsilon}) = C \operatorname{Diag}\left(\sqrt{\Upsilon}\right)$, and then correcting by $-\sum_{i=1}^n x_i \log \gamma_i$.

In what follows, the following notations will be employed:

$$\begin{aligned} &\operatorname{dom}(\Gamma_s) := \{X : X \succeq 0, \operatorname{rank}(X) \ge s\}\,, \text{ and} \\ &\operatorname{dom}(f_{\text{DDFact}};\Upsilon) := \{x : F_{\text{DDFact}}(x;\Upsilon) \in \operatorname{dom}(\Gamma_s)\} \end{aligned}$$

being the domains of $\Gamma_s(X)$ and $f_{\text{DDFact}}(x;\Upsilon)$, respectively. Moreover, we denote

$$\operatorname{dom}(f_{\text{DDFact}};\Upsilon)_+ := \{x : x \ge 0, F_{\text{DDFact}}(x;\Upsilon) \in \operatorname{dom}(\Gamma_s)\}$$

as the intersection of $\operatorname{dom}(f_{\text{DDFact}};\Upsilon)$ and $\mathbb{R}^n_+$. Because the feasible solutions of gscaling-DDFact with finite objective values are evidently confined in $\operatorname{dom}(f_{\text{DDFact}};\Upsilon)_+$, it is enough to concentrate on $\operatorname{dom}(f_{\text{DDFact}};\Upsilon)_+$ instead of $\operatorname{dom}(f_{\text{DDFact}};\Upsilon)$.We wish to highlight the following important point.

Remark. *Generally, we must choose a factorization with k being at least the rank of C, but it is natural to choose one with k equal to the rank of C; for example, via a spectral decomposition of C. In this case, $F_{DDFact}(x;\Upsilon)$ is full-rank if and only if $x \in \mathbb{R}^n_{++}$. In light of this, we can fully understand where on the boundary of the feasible region of gscaling-DDFact, we can encounter solutions not in the interior of* $\operatorname{dom}(f_{DDFact};\Upsilon)_+$.

It is commonly assumed in the literature that the function $f_{\text{DDFact}}(x;\Upsilon)$ may exhibit nonsmooth behavior in x, and toward this end, the supdifferential is characterized. In their work, (Li and Xie, 2023) utilized a Frank-Wolfe algorithm to tackle gscaling-DDFact for the MESP case. Subsequently, (Chen, Fampa, and Lee, 2023) employed a BFGS-based algorithm

of `Knitro` for gscaling-DDFact, to handle both MESP and CMESP, wherein they utilized supgradient information to update the Hessian approximation. This algorithm achieved superior performance in terms of both speed and accuracy, in the spirit of (Lewis and Overton, 2013) which investigated the excellent performance of BFGS on non-smooth problems. In the following section, we will establish that $f_{\text{DDFact}}(x;\Upsilon)$ is actually *in a certain generalized sense* "differentiable" in $x \in \operatorname{dom}(f_{\text{DDFact}};\Upsilon)_+$. These findings serve as a theoretical foundation for the efficiency of algorithms (e.g., those employed by (Chen, Fampa, and Lee, 2023)) that rely on smoothness for their convergence. We will introduce two necessary definitions to facilitate the establishment of our "differentiability" results.

Definition 4.1. *For $x \in \operatorname{dom}(f_{DDFact};\Upsilon)_+$, let the eigenvalues of $F_{DDFact}(x;\Upsilon)$ are $\lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_k = 0$, and $F_{DDFact}(x;\Upsilon) = Q\operatorname{Diag}(\lambda)Q$ with an orthonormal matrix Q. Define $\beta := (\beta_1, \beta_2, \ldots, \beta_k)^\mathsf{T}$ such that*

$$\beta_i := \tfrac{1}{\lambda_i},\ \forall\ i \in [1, \iota],$$
$$\beta_i := \tfrac{s-\iota}{\sum_{i\in[\iota+1,k]}\lambda_i},\ \forall\ i \in [\iota+1, k],$$

where ι is the unique integer defined in Lemma 4.3.

In cases where an explicit analytic formula is unavailable for a function, such as the objective of gscaling-DDFact, the conventional definition of (Fréchet) differentiability only applies to points that exist within the interior of the function domain. This restriction presents challenges when attempting to analyze the properties of a function for points where the conventional definition of (Fréchet) differentiability is not defined, e.g., points at the boundary of the function domain, which is important for understanding the behavior of algorithms having iterates at such points. For our particular function, when we choose a factorization with k equal to the rank of C, such points are precisely the ones with zero components (see Remark 4.4), and might well be visited by active-set methods.[2] To overcome this difficulty, we will extend the definition of (Fréchet) differentiability, in a natural way, to include points at the boundary of a (convex) set.

Definition 4.2. *We define a function $f : \mathbb{R}^n \to \mathbb{R}$ to be generalized differentiable with respect to a set $\mathcal{A} \subseteq \operatorname{dom}(f)$ if a linear operator $g(x) : \mathbb{R}^n \to \mathbb{R}$ exists for all $x \in \mathcal{A}$, such that for all d with $x + d \in \mathcal{A}$, we have $f(x+d) - f(x) - g(x)^\mathsf{T}d = o(\|d\|)$. We refer to $g(x)$ as the generalized gradient with respect to $\mathcal{A}$. We will omit "with respect to $\mathcal{A}$" when it is clear from the context.*

[2] In fact we will see in our computational results (Table 4.4) that they are frequently visited by active-set methods.

Remark. *We would like to highlight that our concept of generalized differentiability is almost as potent as differentiability on $\mathcal{A}$. Specifically, it possesses identical capabilities as differentiability if we use feasible-point optimization algorithms. The reasons are as follows:*

1. *if x lies in the interior of $\mathcal{A}$, then the generalized differentiability and generalized gradient are exactly differentiability and gradient, respectively;*

2. *if x lies on the boundary of $\mathcal{A}$, then the Whitney Extension Theorem guarantees the existence of a compact neighborhood $\mathcal{N}_c(x) \subset \mathcal{A}$ such that the restriction of f on $\mathcal{N}_c(x)$ has a continuously differentiable extension $\hat{f}$ on $\mathbb{R}^n$, with prescribed derivative information on $\mathcal{N}_c(x)$. In other words, $\hat{f}(x) = f(x), \frac{\partial \hat{f}(x)}{\partial x} = g(x)$ for all $x \in \mathcal{N}_c(x)$. Consequently, the generalized differentiability of f is equivalent to its differentiability at the boundary point x, as long as we examine a larger open set that contains a local neighborhood of the boundary point;*

3. *The equation $f(x+d) - f(x) - g(x)^\mathsf{T} d = o(\|d\|)$ implies that as d approaches the zero vector, the expression $f(x+d) - f(x) - g(x)^\mathsf{T} d$ approaches zero, regardless of the* path *taken by d. This statement is essentially the definition of differentiability, except that $x + d \in \mathcal{A}$. Consequently, if an optimization algorithm that always confines its iterates within $\mathcal{A}$ is utilized, the capabilities of generalized differentiability are identical to those of differentiability. Specifically, if this optimization algorithm converges under differentiability, it should also converge under generalized differentiability.*

In the subsequent analysis, we aim to establish the continuous generalized differentiability of the objective of gscaling-DDFact concerning its dependence on $\operatorname{dom}(f_{\text{DDFact}}; \Upsilon)_+$. This will give some theoretical understanding of the good performance of algorithms that empirically have many iterates at the boundary of the feasible region, where smoothness was in question. Such algorithms were observed to outperform interior-point algorithms, which will be shown in the experiments.

Theorem 4.4. *For all $\Upsilon \in \mathbb{R}^n_{++}$ in gscaling-DDFact, the following hold:*

4.4.i. *$z_{DDFact}(\Upsilon)$ yields a valid upper bound for the optimal value of CMESP, i.e., $z(C, s, A, b) \leq z_{DDFact}(\Upsilon)$;*

4.4.ii. *the function $f_{DDFact}(x; \Upsilon)$ is concave in x on $\operatorname{dom}(f_{DDFact}; \Upsilon)_+$;*

4.4.iii. *the function $f_{DDFact}(x; \Upsilon)$ is generalized differentiable with respect to $\operatorname{dom}(f_{DDFact}; \Upsilon)_+$, with generalized gradient*

$$g_x(x; \Upsilon) := \Upsilon \circ \operatorname{diag}\left(FQ \operatorname{Diag}(\beta) Q^\mathsf{T} F^\mathsf{T}\right) - \log \Upsilon,$$

where $C = FF^{\mathsf{T}}$ is a factorization of C and Q, β are defined in Definition 4.1. In particular, $g_x(x;\Upsilon)$ is invariant to different choices of F, Q as long as we change β accordingly;

4.4.iv. given $x \in \operatorname{dom}(f_{DDFact};\Upsilon)_+$, the function $f_{DDFact}(x;\Upsilon)$ is differentiable in Υ with gradient

$$g_\Upsilon(x;\Upsilon) := x \circ \operatorname{diag}\left(FQ \operatorname{Diag}(\beta) Q^{\mathsf{T}} F^{\mathsf{T}}\right) - \operatorname{Diag}(\Upsilon)^{-1} x,$$

where $C = FF^{\mathsf{T}}$ is a factorization of C and Q, β are defined in Definition 4.1. In particular, $g_\Upsilon(x;\Upsilon)$ is invariant to different choices of F, Q, as long as we change β accordingly. Additionally, for MESP, let x^* be an optimal solution to gscaling-DDFact; then we have

$$g_\Upsilon(x^*;\Upsilon)|_{\Upsilon=\mathbf{e}} = 0$$

(which does not *generally hold for CMESP, as we will see in §4.6).*

4.4.v. the function $f_{DDFact}(x;\Upsilon)$ is continuously generalized differentiable in x and continuously differentiable in Υ on $\operatorname{dom}(f_{DDFact};\Upsilon)_+ \times \mathbb{R}^n_{++}$, i.e., $g_x(x;\Upsilon)$ and $g_\Upsilon(x;\Upsilon)$ are continuous on $\operatorname{dom}(f_{DDFact};\Upsilon)_+ \times \mathbb{R}^n_{++}$.

Remark. *(Nikolov, 2015) established Theorem 4.4.*i *for $\Upsilon := \mathbf{e}$, and hence only regarded as a function of x, which was developed further in (Li and Xie, 2023). We generalize this result to the situation where $\Upsilon \in \mathbb{R}^n_{++}$ and is varying. We note that the o-scaled factorization bound for CMESP is invariant under the scale factor (see (Chen, Fampa, and Lee, 2023)), so the use of any type of scaling in the context of the gscaling-DDFact bound is completely new. Theorem 4.4.*ii *is a result of (Nikolov, 2015), with details filled in by (Fampa and Lee, 2022, Section 3.4.2). Theorem 4.4.*iii *is the first differentiablity result of any type for the gscaling-DDFact bound. These results illuminate the success of BFGS-based methods for calculating the gscaling-DDFact bound, not fully anticipated by previous works which exposed only supgradients connected to gscaling-DDFact. Theorem 4.4.*iv *provides the potential for fast algorithms leveraging BFGS-based methods to improve the gscaling-DDFact bound by g-scaling, as we will see in experiments §4.6. These observations and Theorem 4.4.*iv *leave open the interesting question of whether g-scaling can help the gscaling-DDFact bound for MESP; we can interpret Theorem 4.4.*iv *as a partial result toward a negative answer. Theorem 4.4.v is a consequence of Theorems 4.4.iii,iv.*

Proof of Theorem 4.4.i,ii. These are essentially results of (Nikolov, 2015); see also (Fampa

and Lee, 2022, Section 3.4). Intuitively, gscaling-DDFact is the Lagrangian dual of the Lagrangian dual of a nonconvex continuous relaxation of CMESP (see (Chen, Fampa, and Lee, 2023)). Therefore, gscaling-DDFact has a concave objective function, and the optimal value $z_{\text{DDFact}}(\Upsilon)$ serves as valid upper bound for the optimal value of CMESP.

□

Toward establishing the generalized differentiability of $f_{\text{DDFact}}(x;\Upsilon)$, we begin by characterizing the directional derivatives. Toward this end, our first step is to derive the supdifferential of the objective of $\Gamma_s(X)$ with respect to $X \in \text{dom}(\Gamma_s)$.

Proposition 4.5. *(Li and Xie, 2023, Proposition 2) Given $X \in \text{dom}(\Gamma_s)$ with rank $r \in [s,k]$, suppose that its eigenvalues are $\lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_k = 0$ and $X = Q\,\text{Diag}(\lambda)Q^\mathsf{T}$ with an orthonormal matrix Q. Then the supdifferential of the function $\Gamma_s(X)$ at X denoted by $\partial\Gamma_s(X)$ is*

$$\begin{aligned}\partial\Gamma_s(X) = \Big\{ & Q\,\text{Diag}(\beta)Q^\mathsf{T} : X = Q\,\text{Diag}(\lambda)Q^\mathsf{T}, Q \text{ is orthonormal,} \\ & \lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_k = 0, \\ & \beta \in \text{conv}\left\{\beta : \beta_i = \tfrac{1}{\lambda_i}, \forall i \in [\iota], \beta_i = \tfrac{s-\iota}{\sum_{i\in[\iota+1,k]}\lambda_i}, \forall i \in [\iota+1, r], \right. \\ & \left. \beta_i \geq \beta_r, \forall i \in [r+1,k]\right\}\Big\},\end{aligned}$$

where ι is the unique integer defined in Lemma 4.3.

Remark. *If $X \succ 0$, then β is uniquely determined, resulting in a singleton supdifferential $\partial\Gamma_s(X)$ and differentiability of $\Gamma_s(X)$ at X. This further implies the differentiability of $f_{DDFact}(x;\Upsilon)$ in x by the chain rule when $F_{DDFact}(x;\Upsilon) \succ 0$. However, the supdifferential $\partial\Gamma_s(X)$ is not a singleton when X is located on the boundary of the positive-semidefinite cone. This indicates that $\Gamma_s(X)$ is* not *differentiable at such points. However, as we will show later, such non-differentiability does not really transfer to x. In fact, $f_{DDFact}(x;\Upsilon)$ is generalized differentiable at every $x \in \text{dom}\,(f_{DDFact};\Upsilon)_+$. In other words, even if $\Gamma_s(F_{DDFact}(x;\Upsilon))$ is non-differentiable in $F_{DDFact}(x;\Upsilon)$, $f_{DDFact}(x;\Upsilon) = \Gamma_s(F_{DDFact}(x;\Upsilon)) - \sum_{i=1}^n x_i \log\gamma_i$ is still generalized differentiable in x.*

The subsequent step involves computing the directional derivative of $\Gamma_s(X)$ at X, using the supdifferential characterized in Proposition 4.5. It is a well-known fact that if X is located in the interior of $\text{dom}(\Gamma_s)$, then

$$\Gamma_s'(X;D) = \inf_{G\in\partial\Gamma_s(X)} \text{Trace}(G^\mathsf{T}D),$$

where D is a feasible direction at X in $\mathrm{dom}(\Gamma_s)$; see e.g. (Rockafellar, 1997, Theorem 23.4). However, in our current context, X might lie on the boundary of $\mathrm{dom}(\Gamma_s)$. Fortunately, (Moreau, 1966, p. 65) provides a result that ensures that the same formula holds if $\Gamma_s(X)$ is continuous at X. Thus, our first step is to establish the continuity of $\Gamma_s(X)$ at $X \in \mathrm{dom}(\Gamma_s)$.

Lemma 4.6. *$\Gamma_s(X)$ is continuous on its domain.*

Proof. Consider $X \in \mathrm{dom}\,(\Gamma_s)$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq 0$ and ι defined in Lemma 4.3. Let $P \in \mathbb{S}^n$ be such that $X + P \in \mathrm{dom}\,(\Gamma_s)$ and $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_k$ be the eigenvalues of $X + P$ with $\hat{\iota}$ again defined in Lemma 4.3. We will use the continuity of eigenvalues (with respect to entries of the matrix) to prove the result.

We discuss two sub-cases:

1. $\lambda_\iota > \frac{1}{s-\iota}\sum_{\ell=\iota+1}^{k}\lambda_\ell > \lambda_{\iota+1}$. Then for $\|P\|$ small enough, by the continuity of eigenvalues, we have $\hat{\lambda}_\iota > \frac{1}{s-\iota}\sum_{\ell=\iota+1}^{k}\hat{\lambda}_\ell > \hat{\lambda}_{\iota+1}$, which implies $\hat{\iota} = \iota$. Again by the continuity of eigenvalues and $\log(\cdot)$, $\Gamma_s(X)$ is continuous at X on $\mathrm{dom}\,(\Gamma_s)$.

2. $\lambda_\iota > \frac{1}{s-\iota}\sum_{\ell=\iota+1}^{k}\lambda_\ell = \lambda_{\iota+1}$. We have to be more careful in this case as any small $\|P\|$ can make $\hat{\iota}$ different from ι. We first characterize a range where $\hat{\iota}$ should lie within. Let $\iota_e := \max\{i : \lambda_i = \lambda_{\iota+1}, s > i \geq \iota+1\}$. We claim that $\hat{\iota} \in [\iota+1, \iota_e]$. Before proving this claim, we demonstrate three preliminary results:

   (a) For any $i \leq \iota$, $\lambda_i > \frac{1}{s-i}\sum_{\ell=i+1}^{k}\lambda_\ell$;

   (b) For any $i \leq \iota - 1$, $\frac{1}{s-i}\sum_{\ell=i+1}^{k}\lambda_\ell < \lambda_{i+1}$;

   (c) For any $s > i > \iota_e$, $\lambda_i < \frac{1}{s-i}\sum_{\ell=i+1}^{k}\lambda_\ell$.

   Note that (a) holds for $i = 0$. Assume that there exists some $i \leq \iota$ such that $\lambda_i \leq \frac{1}{s-i}\sum_{\ell=i+1}^{k}\lambda_\ell$. Without loss of generality, let i be the minimum integer satisfying this condition. Obviously $i \geq 1$. Furthermore,

$$\frac{1}{s-i+1}\sum_{\ell=i}^{k}\lambda_\ell = \frac{\left(\sum_{\ell=i+1}^{k}\lambda_\ell\right)+\lambda_i}{s-i+1} \geq \frac{1}{s-i+1}((s-i)\lambda_i + \lambda_i) = \lambda_i\,.$$

   By assumption, we also have that $\lambda_{i-1} > \frac{1}{s-i+1}\sum_{\ell=i}^{k}\lambda_\ell$, which together with the above deduction, implies that $\iota = i - 1 \leq \iota - 1$, a contradiction. For (b), if there exists $i \leq \iota - 1$ with $\frac{1}{s-i}\sum_{\ell=i+1}^{k}\lambda_\ell \geq \lambda_{i+1}$, together with (a), we have $\iota = i \leq \iota - 1$, a contradiction. Finally, (c) comes from

$$\begin{aligned}(s-i)\lambda_i < (s-i)\lambda_{\iota+1} &= (s-\iota)\lambda_{\iota+1} - (i-\iota)\lambda_{\iota+1} \\ &\leq \textstyle\sum_{\ell=\iota+1}^{k}\lambda_\ell - (i-\iota)\lambda_{\iota+1}\end{aligned}$$

$$
\begin{aligned}
&= \textstyle\sum_{\ell=\iota+1}^{i}(\lambda_i - \lambda_{\iota+1}) + \sum_{\ell=i+1}^{k}\lambda_\ell \\
&\leq \textstyle\sum_{\ell=i+1}^{k}\lambda_\ell\,.
\end{aligned}
$$

With (a–c), and the continuity of eigenvalues with respect to the entries of a matrix, for $\|P\|$ small enough, we have:

($\hat{a}$) For any $i \leq \iota$, $\hat{\lambda}_i > \frac{1}{s-i}\sum_{\ell=i+1}^{k}\hat{\lambda}_\ell$;

($\hat{b}$) For any $i \leq \iota - 1$, $\frac{1}{s-i}\sum_{\ell=i+1}^{k}\hat{\lambda}_\ell < \hat{\lambda}_{i+1}$;

($\hat{c}$) For any $s > i > \iota_e$, $\hat{\lambda}_i < \frac{1}{s-i}\sum_{\ell=i+1}^{k}\hat{\lambda}_\ell$.

($\hat{a} - \hat{c}$) suggest that $\hat{\iota} \in [\iota+1, \iota_e]$, otherwise the condition $\hat{\lambda}_{\hat{\iota}} > \frac{1}{s-\hat{\iota}}\sum_{\ell=\hat{\iota}+1}^{k}\hat{\lambda}_\ell \geq \hat{\lambda}_{\hat{\iota}+1}$ in Lemma 4.3 will be violated.

From $\frac{1}{s-\iota}\sum_{\ell=\iota+1}^{k}\lambda_\ell = \lambda_{\iota+1}$ and the definition of ι_e, we also have

$$
\begin{aligned}
\lambda_{\iota+1} = \tfrac{1}{s-\iota-1}\textstyle\sum_{\ell=\iota+2}^{k}\lambda_\ell = \lambda_{\iota+2} &= \tfrac{1}{s-\iota-2}\textstyle\sum_{\ell=\iota+3}^{k}\lambda_\ell \\
&= \cdots \\
&= \lambda_{\iota_e} = \tfrac{1}{s-\iota_e}\textstyle\sum_{\ell=\iota_e+1}^{k}\lambda_\ell\,. \qquad (4.5)
\end{aligned}
$$

We are now ready to prove the continuity results. Because $\hat{\iota} \in [\iota+1, \iota_e]$ and (4.5), we have

$$
\begin{aligned}
&\textstyle\sum_{\ell=1}^{\iota}\log(\lambda_\ell) + (s-\iota)\log\left(\frac{1}{s-\iota}\sum_{\ell=\iota+1}^{k}\lambda_\ell\right) \\
&\quad= \textstyle\sum_{\ell=1}^{\iota}\log(\lambda_\ell) + (\hat{\iota}-\iota)\log\left(\frac{1}{s-\iota}\sum_{\ell=\iota+1}^{k}\lambda_\ell\right) + (s-\hat{\iota})\log\left(\frac{1}{s-\iota}\sum_{\ell=\iota+1}^{k}\lambda_\ell\right) \\
&\quad= \textstyle\sum_{\ell=1}^{\iota}\log(\lambda_\ell) + \sum_{\ell=\iota+1}^{\hat{\iota}}\log(\lambda_\ell) + (s-\hat{\iota})\log\left(\frac{1}{s-\hat{\iota}}\sum_{\ell=\hat{\iota}+1}^{k}\lambda_\ell\right) \\
&\quad= \textstyle\sum_{\ell=1}^{\hat{\iota}}\log(\lambda_\ell) + (s-\hat{\iota})\log\left(\frac{1}{s-\hat{\iota}}\sum_{\ell=\hat{\iota}+1}^{k}\lambda_\ell\right).
\end{aligned}
$$

Note that the above equation holds for any $\|P\|$ small enough, and with its corresponding $\hat{\iota}$ and at $X+P$, we have

$$
\Gamma_s(X+P) = \textstyle\sum_{\ell=1}^{\hat{\iota}}\log(\hat{\lambda}_\ell) + (s-\hat{\iota})\log\left(\frac{1}{s-\hat{\iota}}\sum_{\ell=\hat{\iota}+1}^{k}\hat{\lambda}_\ell\right).
$$

Then by the continuity of eigenvalues (with respect to elements of the matrix) and $\log(\cdot)$ function, we conclude that $\Gamma_s(X)$ is continuous on $\mathrm{dom}(\Gamma_s)$.

□

The continuity of $\Gamma_s(X)$ in X together with the continuity of eigenvalues in matrix elements also implies the following.

Corollary 4.7. *$f_{DDFact}(x;\Upsilon)$ is continuous in (x,Υ) on $\operatorname{dom}(f_{DDFact};\Upsilon)_+$.*

Utilizing the above results, we can characterize the directional derivative of $\Gamma_s(X)$ and further characterize the directional derivative of $f_{\mathrm{DDFact}}(x;\Upsilon)$.

Proposition 4.8. *For $X \in \operatorname{dom}(\Gamma_s)$, let $D \in \mathbb{S}^n$ be such that $X + D \in \operatorname{dom}(\Gamma_s)$; then the directional derivative of $\Gamma_s(X)$ at X in the direction D, denoted $\Gamma_s'(X;D)$, exists and*

$$\Gamma_s'(X;D) = \inf_{G \in \partial\Gamma_s(X)} \operatorname{Trace}(G^{\mathsf{T}} D).$$

Proof. By definition, (Li and Xie, 2023, Lemma 3) and Lemma 4.6, $\Gamma_s(X)$ is convex, finite, and continuous in X over $\operatorname{dom}(\Gamma_s)$. Then the conclusion follows by (Moreau, 1966, p. 65). □

Proposition 4.9. *For $x \in \operatorname{dom}(f_{DDFact};\Upsilon)_+$, let $d \in \mathbb{R}^n$ be such that $x + d \in \operatorname{dom}(f_{DDFact};\Upsilon)_+$; then the directional derivative of $f_{DDFact}(x;\Upsilon)$ at x in the direction d exists, and*

$$f'_{DDFact}(x;\Upsilon;d) = \left(\Upsilon \circ \operatorname{diag}\left(FQ \operatorname{Diag}(\beta) QF^{\mathsf{T}}\right)\right)^{\mathsf{T}} d - \log(\Upsilon)^{\mathsf{T}} d,$$

where $C = FF^{\mathsf{T}}$ is a factorization of C, and Q, β are defined in Definition 4.1. In particular, $FQ \operatorname{Diag}(\beta) QF^{\mathsf{T}}$ and thus $f'_{DDFact}(x;\Upsilon;d)$ is invariant to the choice of F, Q, as long as we change β accordingly.

Proof. By Theorem 4.4.ii, we have that $f_{\mathrm{DDFact}}(x;\Upsilon)$ is concave. Then by (Rockafellar, 1997, Theorem 23.1), the directional derivative $f'_{\mathrm{DDFact}}(x;\Upsilon;d)$ exists and

$$\begin{aligned}
& f'_{\mathrm{DDFact}}(x;\Upsilon;d) \\
&= \lim_{t\to 0^+} \tfrac{f_{\mathrm{DDFact}}(x+td;\Upsilon) - f_{\mathrm{DDFact}}(x;\Upsilon)}{t} \\
&= \lim_{t\to 0^+} \tfrac{\Gamma_s(F_{\mathrm{DDFact}}(x+td;\Upsilon)) - \Gamma_s(F_{\mathrm{DDFact}}(x;\Upsilon)) + t\log(\Upsilon)^{\mathsf{T}} d}{t} \\
&= \lim_{t\to 0^+} \tfrac{\Gamma_s(F_{\mathrm{DDFact}}(x;\Upsilon) + tF_{\mathrm{DDFact}}(d;\Upsilon)) - \Gamma_s(F_{\mathrm{DDFact}}(x;\Upsilon)) + t\log(\Upsilon)^{\mathsf{T}} d}{t} \\
&= \Gamma_s'\left(F_{\mathrm{DDFact}}(x;\Upsilon); F_{\mathrm{DDFact}}(d;\Upsilon)\right) + \log(\Upsilon)^{\mathsf{T}} d \\
&= \inf_{G \in \partial\Gamma_s(F_{\mathrm{DDFact}}(x;\Upsilon))} \operatorname{Trace}\left(G^{\mathsf{T}} F_{\mathrm{DDFact}}(d;\Upsilon)\right) + \log(\Upsilon)^{\mathsf{T}} d,
\end{aligned}$$

where the last equation is due to Proposition 4.8. Let $\Theta(x,\Upsilon)$ denote the set of (Q,β) in

the characterization of $\partial\Gamma_s\left(F_{\text{DDFact}}(x;\Upsilon)\right)$, as described in Proposition 4.5. In particular,

$$\begin{aligned}\Theta(x,\Upsilon) = \Big\{ & (Q,\beta) : F_{\text{DDFact}}(x;\Upsilon) = Q\,\text{Diag}(\lambda)Q^{\mathsf{T}}, Q \text{ is orthonormal},\\ & \lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_k = 0,\\ & \beta \in \text{conv}\Big\{\beta : \beta_i = \tfrac{1}{\lambda_i}, \forall i \in [\iota], \beta_i = \tfrac{s-\iota}{\sum_{i\in[\iota+1,k]}\lambda_i}, \forall i \in [\iota+1, r],\\ & \beta_i \geq \beta_r, \forall i \in [r+1,k]\Big\}\Big\},\end{aligned}$$

where ι is the unique integer defined in Lemma 4.3. According to the former derivation,

$$\begin{aligned} & f'_{\text{DDFact}}(x;\Upsilon;d)\\ = & \inf_{G\in\partial\Gamma_s(F_{\text{DDFact}}(x;\Upsilon))} \text{Trace}\left(G^{\mathsf{T}}F_{\text{DDFact}}(d;\Upsilon)\right) + \log(\Upsilon)^{\mathsf{T}}d\\ = & \inf_{(Q,\beta)\in\Theta(x;\Upsilon)} \text{Trace}\left(FQ\,\text{Diag}(\beta)Q^{\mathsf{T}}F^{\mathsf{T}}\,\text{Diag}(\Upsilon\circ d)\right) + \log(\Upsilon)^{\mathsf{T}}d\\ = & \inf_{(Q,\beta)\in\Theta(x;\Upsilon)} \left(\Upsilon\circ\text{diag}\left(FQ\,\text{Diag}(\beta)Q^{\mathsf{T}}F^{\mathsf{T}}\right)\right)^{\mathsf{T}} d + \log(\Upsilon)^{\mathsf{T}}d. \end{aligned} \tag{4.6}$$

We now show that the infimum of (4.6) is obtained by (Q,β) characterized in Definition 4.1. For simplicity, we let

$$\begin{aligned} g(Q,\beta;\Upsilon) := & \ \Upsilon\circ\text{diag}\left(FQ\,\text{Diag}(\beta)Q^{\mathsf{T}}F^{\mathsf{T}}\right);\\ g_j(Q,\beta;\Upsilon) := & \ \gamma_i F_{i\cdot}Q\,\text{Diag}(\beta)Q^{\mathsf{T}}F_{i\cdot}^{\mathsf{T}}, && \text{the } i^{\text{th}} \text{ element of } g(Q,\beta;\Upsilon);\\ q_j := & \ Q_{\cdot j}, && \text{the } j^{\text{th}} \text{ column of } Q \end{aligned} \tag{4.7}$$

(where $F_{i\cdot}$ denotes the i^{th} row of F). By the definition of $\Theta(x,\Upsilon)$, we also have that if $j_1 > j_2$, the eigenvalues $\lambda_{j_1}, \lambda_{j_2}$ associated with q_{j_1}, q_{j_2} satisfy $\lambda_{j_1} \leq \lambda_{j_2}$.

We claim that for any $1 \leq i \leq n, r < j \leq k$ where $x_i > 0$, $F_{i\cdot}q_j = 0$. First by the characterization of Q in $\Theta(x,\Upsilon)$, we have that $q_{j_1}^{\mathsf{T}}q_{j_2} = 0$ for all $1 \leq j_1 \leq r < j_2 \leq k$, because q_{j_1}, q_{j_2} lie in eigenspaces corresponding to different eigenvalues. Therefore, it is enough to prove that $F_{i\cdot}$ lies in the space spanned by $\{q_j \ : \ 1 \leq j \leq r\}$. Notice that $F_{\text{DDFact}}(x;\Upsilon) = \sum_{i=1}^n \gamma_i x_i F_{i\cdot}^{\mathsf{T}}F_{i\cdot} = F^{\mathsf{T}}\,\text{Diag}(\Upsilon\circ x)F$. Therefore, the column space of $F_{\text{DDFact}}(x;\Upsilon)$ is equal to the row space of $\text{Diag}(\Upsilon\circ x)^{\frac{1}{2}}F$, which is in turn equal to the space spanned by $\{F_{i\cdot}^{\mathsf{T}} \ : \ 1 \leq i \leq n,\, x_i > 0\}$. On the other hand, $F_{\text{DDFact}}(x;\Upsilon) = Q\,\text{Diag}(\lambda)Q^{\mathsf{T}}$, and thus the column space of $F_{\text{DDFact}}(x;\Upsilon)$ is equal to the row space of $\text{Diag}(\lambda)^{\frac{1}{2}}Q^{\mathsf{T}}$, which is in turn equal to the space spanned by $\{q_j \ : \ 1 \leq j \leq r\}$. Therefore, we have proved that

$$\text{span}\{F_{i\cdot}^{\mathsf{T}} \ : \ 1 \leq i \leq n,\, x_i > 0\} = \text{span}\{q_j \ : \ 1 \leq j \leq r\},$$

which implies that for all $1 \le i \le n, r < j \le k$ where $x_i > 0$, $F_{i\cdot} q_j = 0$. With this result, we have when $x_i > 0$,

$$g_i(Q, \beta; \Upsilon) = \gamma_i F_{i\cdot} Q \operatorname{Diag}(\beta) Q^\mathsf{T} F_{i\cdot}^\mathsf{T} = \gamma_i \textstyle\sum_{j=1}^r \beta_j \|F_{i\cdot} q_j\|^2,$$

which is invariant to $(Q, \beta) \in \Theta(x; \Upsilon)$ because $\beta_j, 1 \le j \le r$ are fixed and Q is not contained in the right-hand side formula.

We choose some $(\hat{Q}, \hat{\beta})$ defined in Definition 4.1. Note that the choice of $(\hat{Q}, \hat{\beta})$ is not unique. Then we can write the directional derivative as

$$\begin{aligned}
&f'_{\text{DDFact}}(x; \Upsilon; d) \\
&= \inf_{(Q,\beta)\in\Theta(x;\Upsilon)} \textstyle\sum_{x_i>0} g_i(Q, \beta; \Upsilon) d_i + \sum_{x_i=0} g_i(Q, \beta; \Upsilon) d_i + \log(\Upsilon)^\mathsf{T} d \\
&= \textstyle\sum_{x_i>0} g_i\left(\hat{Q}, \hat{\beta}; \Upsilon\right) d_i + \inf_{(Q,\beta)\in\Theta(x;\Upsilon)} \sum_{x_i=0} \gamma_i \sum_{j=1}^n \beta_j \|F_{i\cdot} q_j\|^2 d_i + \log(\Upsilon)^\mathsf{T} d \\
&= \textstyle\sum_{x_i>0} g_i\left(\hat{Q}, \hat{\beta}; \Upsilon\right) d_i + \inf_{(Q,\beta)\in\Theta(x;\Upsilon)} \sum_{j=1}^n \beta_j \sum_{x_i=0} \gamma_i \|F_{i\cdot} q_j\|^2 d_i + \log(\Upsilon)^\mathsf{T} d.
\end{aligned}$$

Note that if $x_i = 0$, we must have $d_i \ge 0$ to make $x + d \in \operatorname{dom}(f_{\text{DDFact}}; \Upsilon)_+$. Therefore, for each $r < j \le k$, $\sum_{i:x_i=0} \gamma_i \|F_{i\cdot} q_j\| d_i \ge 0$, and the infimum is achieved if and only if each $\beta_j, r < j \le k$ takes the minimum value in $\Theta(x; \Upsilon)$, which is easy to see that is just the value in Definition 4.1. In particular, $(\hat{Q}, \hat{\beta})$ is such a choice. Specifically, we have

$$\begin{aligned}
&f'_{\text{DDFact}}(x; \Upsilon; d) \\
&\quad = \textstyle\sum_{x_i>0} g_i\left(\hat{Q}, \hat{\beta}; \Upsilon\right) d_i + \gamma_i \sum_{j=1}^n \hat{\beta}_j \sum_{x_i=0} \|F_{i\cdot} \hat{q}_j\|^2 d_i + \log(\Upsilon)^\mathsf{T} d \\
&\quad = \textstyle\sum_{x_i>0} g_i\left(\hat{Q}, \hat{\beta}; \Upsilon\right) d_i + \gamma_i \sum_{x_i=0} g_i\left(\hat{Q}, \hat{\beta}; \Upsilon\right) d_i + \log(\Upsilon)^\mathsf{T} d \\
&\quad = g_i\left(\hat{Q}, \hat{\beta}; \Upsilon\right)^\mathsf{T} d + \log(\Upsilon)^\mathsf{T} d.
\end{aligned}$$

Note that $\hat{Q}$ can be any Q defined in Definition 4.1, and the value of $g(Q, \beta; \Upsilon)$ is invariant as long as we change $\hat{\beta}$ accordingly. The invariance relative to F is due to the invariance of $f_{\text{DDFact}}(x; \Upsilon)$ relative to F (see (Chen, Fampa, and Lee, 2023, Theorem 2.2)). □

With the characterization of directional derivative in Proposition 4.9, we can prove the general differentiability with respect to $\operatorname{dom}(f_{\text{DDFact}}; \Upsilon)_+$ as defined in Definition 4.2.

Proof of Theorem 4.4 iii, iv, and v. We continue the proof of Theorem 4.4 here.

4.4.iii: By Proposition 4.9, let $g_x(x; \Upsilon) := \Upsilon \circ \operatorname{diag}\left(FQ \operatorname{Diag}(\beta) Q F^\mathsf{T}\right) + \log(\Upsilon)$ for any (Q, β) defined in Definition 4.1. Proposition 4.9 shows that $(x; \Upsilon)$ is invariant to the choice

of (Q, β) and F. Then the directional derivative of $f_{\text{DDFact}}(x; \Upsilon)$ with respect to $x \in \text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$ and feasible direction $d \in \mathbb{R}^n$ such that $x + d \in \text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$ is $g_x(x; \Upsilon)^\mathsf{T} d$.

We first demonstrate two preliminary results:

(a) Given $x \in \text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$, we define the neighbourhood of x with radius r with respect to $\text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$ as

$$\mathcal{N}_r(x) := \left\{ y \ : \ \|y - x\| \leq r, y \in \text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+ \right\}.$$

We claim that for r small enough, $\mathcal{N}_r(x)$ is a compact set. Recall that:

$$\begin{aligned} &\text{dom}\,(\Gamma_s) := \{X : X \succeq 0, \text{rank}(X) \geq s\}, \text{ and} \\ &\text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+ := \{x : x \geq 0, F_{\text{DDFact}}(x; \Upsilon) \in \text{dom}\,(\Gamma_s)\}. \end{aligned}$$

By the continuity of eigenvalues, there is some small enough $\tilde{r} > 0$ such that when $r \leq \tilde{r}$, $F_{\text{DDFact}}(y; \Upsilon)$ has at least the same number of nonzero eigenvalues as $F_{\text{DDFact}}(x; \Upsilon)$, and so $\text{rank}\,(F_{\text{DDFact}}(y; \Upsilon)) \geq s$. Moreover, note that the set $\{x \ : \ F_{\text{DDFact}}(x; \Upsilon) \succeq 0\}$ and the non-negative cone $\mathbb{R}^n_+ = \{x \ : \ x \geq 0\}$ are closed. So $\mathcal{N}_r(x)$ can be seen as the intersection of $\{x \ : \ F_{\text{DDFact}}(x; \Upsilon) \succeq 0\}$, $\mathbb{R}^n_+$, and the sphere $\{y \ : \ \|y - x\| \leq r\}$, thus is closed and bounded, and thus compact. Furthermore, we have shown in Corollary 4.7 that $f_{\text{DDFact}}(x; \Upsilon)$ is continuous over $\text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$, thus uniform continuous over $\mathcal{N}_r(x)$ for $r \leq \tilde{r}$.

(b) Given $x \in \text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$, we define the circle of x with radius r with respect to $\text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$ as

$$\mathcal{C}_r(x) := \left\{ y \ : \ \|y - x\| = r, y \in \text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+ \right\}.$$

With similar logic to the above, when $r \leq \tilde{r}$, $\mathcal{C}_r(x)$ is closed and bounded. Then by Heine-Borel Theorem, for any $\epsilon > 0$, there exists a finite set $F \subset \mathcal{C}_{\tilde{r}}(x)$ such that for any $y \in \mathcal{C}_{\tilde{r}}(x)$, there exists $u \in F$ such that $\|y - u\| < \epsilon$.

Now we are ready to establish generalized differentiability of $f_{\text{DDFact}}(x; \Upsilon)$ with respect to $\text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$. In particular, we want to demonstrate that for any $\epsilon > 0$, there exists some $\delta > 0$ such that whenever $y \in \text{dom}\,(f_{\text{DDFact}}; \Upsilon)_+$ and $\|y - x\| < \delta$, we have

$$\left| f_{\text{DDFact}}(y; \Upsilon) - f_{\text{DDFact}}(x; \Upsilon) - g_x(x; \Upsilon)^\mathsf{T}(y - x) \right| < \epsilon.$$

We will assume that $g_x(x;\Upsilon) \neq 0$, because the case where $g_x(x;\Upsilon) = 0$ is implied by the continuity of $f_{\text{DDFact}}(x;\Upsilon)$ (see Corollary 4.7). We have the following four facts:

(1) from (a), $f_{\text{DDFact}}(x;\Upsilon)$ is uniformly continuous on $\mathcal{N}_{\tilde{r}}(x)$. Then given $\epsilon > 0$, there is some $\delta_1 > 0$ such that for any $x_1, x_2 \in \mathcal{N}_{\tilde{r}}(x)$ such that $\|x_1 - x_2\| < \frac{\delta_1}{\|g_x(x;\Upsilon)\|}$, we have $|f_{\text{DDFact}}(x_1;\Upsilon) - f_{\text{DDFact}}(x_2;\Upsilon)| < \frac{\epsilon}{3}$.

(2) given $\delta_1 > 0$, by (b), there is some finite set $F \subset \mathcal{C}_{\tilde{r}}(x)$ such that for every $y \in \mathcal{C}_{\tilde{r}}(x)$, there exists $u \in F$ such that $\|y - u\| < \frac{\min\{\epsilon,\delta_1\}}{3\cdot\|g_x(x;\Upsilon)\|}$.

(3) because of the finiteness of F and the existence of the directional derivative in direction $u - x$, $\forall\, u \in F$, given $\epsilon > 0$, there exists some $\delta_2 \leq 1$ such that for any $u \in F$, when $t < \delta_2$, we have

$$\left|f_{\text{DDFact}}(x + t(u-x);\Upsilon) - f_{\text{DDFact}}(x;\Upsilon) - t g_x(x;\Upsilon)^{\mathsf{T}}(u-x)\right| < \tfrac{\epsilon}{3}.$$

Note that $t < \delta_2$ is equivalent to that $t \cdot \|u - x\| < \delta_3 := \delta_2 \cdot \tilde{r}$.

(4) for every $y \in \mathcal{N}_{\tilde{r}}(x)$, we have that $x + \frac{y-x}{\|y-x\|} \cdot \tilde{r} \in \mathcal{C}_{\tilde{r}}(x)$. By (2), there is some $u \in F$ such that

$$\left\|x + \tfrac{y-x}{\|y-x\|} \cdot \tilde{r} - u\right\| < \tfrac{\min\{\epsilon,\delta_1\}}{3\cdot\|g_x(x;\Upsilon)\|},$$

and thus

$$\left\|y - \left(x + \tfrac{u-x}{\tilde{r}} \cdot \|y-x\|\right)\right\| = \tfrac{\|y-x\|}{\tilde{r}} \cdot \left\|x + \tfrac{y-x}{\|y-x\|} \cdot \tilde{r} - u\right\| < \tfrac{\min\{\epsilon,\delta_1\}}{3\cdot\|g_x(x;\Upsilon)\|}.$$

From (1–4), given $\epsilon > 0$, there exists $\delta_3 > 0$ and a finite set $F \subset \mathcal{C}_{\tilde{r}}(x)$ such that for any $y \in \operatorname{dom}(f_{\text{DDFact}};\Upsilon)_+$ and $\|y - x\| < \delta_3$, there exists some $u \in F$ such that

$$\begin{aligned}
&\left|f_{\text{DDFact}}(y;\Upsilon) - f_{\text{DDFact}}(x;\Upsilon) - g_x(x;\Upsilon)^{\mathsf{T}}(y-x)\right| \\
&\quad\leq |f_{\text{DDFact}}(y;\Upsilon) - f_{\text{DDFact}}(\hat{y};\Upsilon)| \\
&\qquad + \left|f_{\text{DDFact}}(\hat{y};\Upsilon) - f_{\text{DDFact}}(x;\Upsilon) - g_x(x;\Upsilon)^{\mathsf{T}}(\hat{y}-x)\right| \\
&\qquad + \left|g_x(x;\Upsilon)^{\mathsf{T}}(y-\hat{y})\right| \\
&\quad< \frac{\epsilon}{3} + \frac{\epsilon}{3} + \|g_x(x;\Upsilon)\| \frac{\min\{\epsilon,\delta_1\}}{3\|g_x(x;\Upsilon)\|} \\
&\quad< \epsilon,
\end{aligned}$$

where $\hat{y} = x + \frac{u-x}{\tilde{r}} \|y - x\|$. Finally, the invariance of $g_x(x;\Upsilon)$ to F, Q as long as we change β accordingly follows from Proposition 4.9.

4.4.iv: For the first part, note that $F_{\text{DDFact}}(x;\Upsilon) = \sum_{i=1}^{n} \gamma_i x_i F_{i\cdot}^{\mathsf{T}} F_{i\cdot}$, and for every $x \in$

$\operatorname{dom}(f_{\text{DDFact}};\Upsilon)_+$ and $\Upsilon \in \mathbb{R}^n_{++}$, $F_{\text{DDFact}}(x;\Upsilon) \in \operatorname{dom}(\Gamma_s)$, and is thus well defined. On the other hand, by switching the value of x and Υ, we find that $F_{\text{DDFact}}(\Upsilon;x) = F_{\text{DDFact}}(x;\Upsilon)$. We can use the same method which we used to derive the generalized gradient with respect to x to derive the generalized gradient with respect to Υ. This means that $f_{\text{DDFact}}(x;\Upsilon)$ is generalized differentiable at Υ with generalized gradient

$$g_\Upsilon(x;\Upsilon) = x \circ \operatorname{diag}\left(FQ \operatorname{Diag}(\beta) Q^\mathsf{T} F^\mathsf{T}\right) - \operatorname{Diag}(\Upsilon)^{-1} x,$$

where $C = FF^\mathsf{T}$ is a factorization of C and (Q,β) are defined in Definition 4.1. In particular, $g_\Upsilon(x;\Upsilon)$ is invariant to different choices of F, Q and thus well defined. Moreover, because $\Upsilon \in \mathbb{R}^n_{++}$ lies in the interior of $\mathbb{R}^n_{++}$, the generalized differentiability reduces to differentiability. Moreover, the invariance of $g_\Upsilon(x;\Upsilon)$ to F, Q as long as we change β accordingly follows the same logic as Theorem 4.4.iii.

For the second part, note that $g_\Upsilon(x^*;\Upsilon)|_{\Upsilon=\mathbf{e}} = 0$ is equivalent to $x^* \circ (g^*(Q;\beta;\mathbf{e}) - \mathbf{e}) = 0$, where $g^*(Q;\beta;\mathbf{e}) = \operatorname{diag}\left(FQ \operatorname{Diag}(\beta) Q^\mathsf{T} F^\mathsf{T}\right)$ as defined in (4.7), specifically for x^*. This is further equivalent to

$$g^*_i(Q;\beta;\mathbf{e}) = 1,\ \forall x^*_i > 0.$$

In the following proof, we will leverage the KKT conditions of gscaling-DDFact which we present here: for any optimal solution x^* to gscaling-DDFact, there is some $\upsilon^* \in \mathbb{R}^n, \nu^* \in \mathbb{R}^n, \pi^* \in \mathbb{R}^m$ such that

$$\begin{aligned}
&\mathbf{e}^\mathsf{T} x^* = s,\ Ax^* \le b,\ 0 \le x^* \le \mathbf{e},\\
&\upsilon^* \ge 0,\ \nu^* \ge 0,\ \pi^* \ge 0,\\
&g^*(Q,\beta;\Upsilon) + \upsilon^* - \nu^* - A^\mathsf{T}\pi^* - \tau^*\mathbf{e} = 0,\\
&\pi^* \circ (b - Ax^*) = 0,\ \upsilon^* \circ x^* = 0,\ \nu^* \circ (\mathbf{e} - x^*) = 0,
\end{aligned} \tag{DDFact-KKT}$$

where $g^*(Q,\beta;\Upsilon) = \Upsilon \circ \operatorname{diag}\left(FQ \operatorname{Diag}(\beta) Q^\mathsf{T} F^\mathsf{T}\right)$ as defined in (4.7), specifically for x^*. The existence of $\upsilon^* \in \mathbb{R}^n, \nu^* \in \mathbb{R}^n, \tau^* \in \mathbb{R}, \pi^* \in \mathbb{R}^m$ is due to that: (1) gscaling-DDFact is a generalized differentiable convex-optimization problem; (2) Slater's condition holds because of the affine constraints describing the feasible region of gscaling-DDFact.

By (Li and Xie, 2023, Section 3.1), when $\Upsilon = \mathbf{e}$ and there are not linear contraints $Ax \le b$, then there is a closed-form solution $(\upsilon^*, \nu^*, \tau^*)$ to DDFact-KKT given x^*. Suppose that σ is a permutation of $\{1, 2, \cdots, n\}$ such that

$$(g^*(Q;\beta;\mathbf{e}))_{\sigma(1)} \ge (g^*(Q;\beta;\mathbf{e}))_{\sigma(2)} \ge \cdots \ge (g^*(Q;\beta;\mathbf{e}))_{\sigma(n)},$$

where $(g^*(Q;\beta;\mathbf{e}))_i$ denotes the i^{th} element of $g^*(Q;\beta;\mathbf{e})$. Then

$$\begin{aligned}
&\tau^* = (g^*(Q;\beta;\mathbf{e}))_{\sigma(s)}\,,\\
&\nu^*_{\sigma(i)} = \begin{cases} (g^*(Q;\beta;\mathbf{e}))_{\sigma(i)} - \tau^*, & \forall\ 1 \le i \le s;\\ 0, & \forall\ s+1 \le i \le n, \end{cases}\\
&\upsilon^* = \nu^* + \tau^*\mathbf{e} - g^*(Q;\beta;\mathbf{e}).
\end{aligned}$$

We claim that

$$\textstyle\sum_{i\in\{1,2,\ldots,n\}} x^*_{\sigma(i)}\,(g^*(Q;\beta;\mathbf{e}))_{\sigma(i)} = \sum_{i\in\{1,2,\ldots,s\}} (g^*(Q;\beta;\mathbf{e}))_{\sigma(i)} = s.$$

In fact, by DDFact-KKT, we have

$$\begin{aligned}
0 &= x^* \circ (g^*(Q;\beta;\mathbf{e}) + \upsilon^* - \nu^* - \tau^*\mathbf{e})\\
&= x^* \circ g^*(Q;\beta;\mathbf{e}) + x^* \circ \upsilon^* - x^* \circ \nu^* - \tau^* x^*\\
&= x^* \circ g^*(Q;\beta;\mathbf{e}) - x^* \circ \nu^* - \tau^* x^*\\
&= x^* \circ g^*(Q;\beta;\mathbf{e}) - \nu^* - \tau^* x^*,
\end{aligned}$$

and further

$$\begin{aligned}
0 &= \mathbf{e}^{\mathsf{T}}\,(x^* \circ g^*(Q;\beta;\mathbf{e}) - \nu^* - \tau^* x^*)\\
&= \textstyle\sum_{i\in\{1,2,\ldots,n\}} x^*_{\sigma(i)}\,(g^*(Q;\beta;\mathbf{e}))_{\sigma(i)} + \sum_{i\in\{1,2,\ldots,s\}} \nu^*_{\sigma(i)} + \tau^* s\\
&= \textstyle\sum_{i\in\{1,2,\ldots,n\}} x^*_{\sigma(i)}\,(g^*(Q;\beta;\mathbf{e}))_{\sigma(i)} - \sum_{i\in\{1,2,\ldots,s\}} (g^*(Q;\beta;\mathbf{e}))_{\sigma(i)}\,.
\end{aligned}$$

On the other hand, by (Chen, Fampa, and Lee, 2023), the duality gap of gscaling-DDFact is $\mathbf{e}^{\mathsf{T}}\nu^* + \tau^* s - s = 0$ and thus

$$\sum_{i\in\{1,2,\ldots,n\}} x^*_{\sigma(i)}\,(g^*(Q;\beta;\mathbf{e}))_{\sigma(i)} = \sum_{i\in\{1,2,\ldots,s\}} (g^*(Q;\beta;\mathbf{e}))_{\sigma(i)} = s. \tag{4.8}$$

Furthermore, we claim that if $x^*_{\sigma(i)} = 1$, then $g_{\sigma(i)}(x^*) \le 1$. Note that by the proof of Proposition 4.9, letting q_j be the j^{th} column of Q, we have

$$\begin{aligned}
(g(Q;\beta;\mathbf{e}))_{\sigma(i)} &= F_{\sigma(i)\cdot} Q\,\mathrm{Diag}\,(\beta)\,Q^{\mathsf{T}} F^{\mathsf{T}}_{\sigma(i)\cdot}\\
&= \textstyle\sum_{j=1}^{r} \beta_j F_{\sigma(i)\cdot} q_j q_j^{\mathsf{T}} F_{\sigma(i)\cdot}\\
&\le \textstyle\sum_{j=1}^{r} \frac{1}{\lambda_j} F_{\sigma(i)\cdot} q_j q_j^{\mathsf{T}} F_{\sigma(i)\cdot}
\end{aligned}$$

$$
\begin{aligned}
&= F_{\sigma(i)\cdot} \left(F_{\text{DDFact}}(x;\mathbf{e})\right)^{\dagger} F_{\sigma(i)\cdot}^{\mathsf{T}} \\
&= F_{\sigma(i)\cdot} \left(F_{\sigma(i)\cdot}^{\mathsf{T}} F_{\sigma(i)\cdot} + \textstyle\sum_{j\neq\sigma(i)} x_j^* F_{j\cdot}^{\mathsf{T}} F_{j\cdot}\right)^{\dagger} F_{\sigma(i)\cdot}^{\mathsf{T}} \\
&\leq F_{\sigma(i)\cdot} \left(F_{\sigma(i)\cdot}^{\mathsf{T}} F_{\sigma(i)\cdot}\right)^{\dagger} F_{\sigma(i)\cdot}^{\mathsf{T}} = 1, \qquad (4.9)
\end{aligned}
$$

where the first inequality is due to Lemma 4.3 and Definition 4.1, and the second inequality is due to the Sherman–Morrison formula for the Moore-Penrose inverse. (4.8) and (4.9) together imply that

$$
\left(g(Q;\beta;\mathbf{e})\right)_{\sigma(1)} = \cdots = \left(g(Q;\beta;\mathbf{e})\right)_{\sigma(s)} = 1.
$$

Moreover, for $i > s$ such that $x^*_{\sigma(i)} > 0$, we must have $\left(g(Q;\beta;\mathbf{e})\right)_{\sigma(i)} = \left(g(Q;\beta;\mathbf{e})\right)_{\sigma(s)}$, otherwise we contradict (4.8), due to the non-increasingness of $\left(g(Q;\beta;\mathbf{e})\right)_{\sigma(i)}$ in i.

4.4.v: By the generalized gradients characterized for x and gradients characterized for ψ in Theorem 4.4.iii,iv, we only need to prove the continuity of $g(Q;\beta;\Upsilon)$ as defined in (4.7) with respect to (x,Υ). Because of the invariance of $g(Q;\beta;\Upsilon)$ to F, Q as long as we change β accordingly, we can fix F, Q; then the conclusion follows from the continuity of eigenvalues in the matrix elements.

□

4.5 Computing optimal g-scaling parameters

In this section, we discuss our algorithms for determining optimal g-scaling vectors for gscaling-BQP and gscaling-linx, as well as for the selection of *good* g-scaling vectors for gscaling-DDFact. For gscaling-DDFact, we can only aim for *good*, because of the lack of a convexity result concerning the g-scaling vector for gscaling-DDFact; despite this, results presented in §4.6 demonstrate that, in many cases, the gscaling-DDFact bounds computed with the best g-scaling vectors obtained are the strongest that we have, demonstrating the effectiveness of such algorithms.

For notational generality, we consider an upper-bound form for CMESP, which encompasses gscaling-BQP, gscaling-linx, and gscaling-DDFact as particular instantiations. Specifically, we define a general upper bound for CMESP of the form:

$$
\begin{aligned}
z(\psi) \;:=\; \max \quad & f(x;\psi) \\
\text{s.t.} \quad & g_i(x) \leq 0,\ \forall i = 1,2,\cdots,m_1\,; \\
& h_j(x) = 0,\ \forall j = 1,2,\cdots,m_2\,,
\end{aligned}
\qquad \text{(CMESP-UB)}
$$

where $f : \mathrm{dom}(f) \to \mathbb{R}$, $g_i : \mathrm{dom}(g_i) \to \mathbb{R}$, and $h_j : \mathrm{dom}(h_j) \to \mathbb{R}$ (with the data C, s, A, b being absorbed into these functions). We assume that T, the set of possible values for the parameter vector ψ to be open. We also assume that $f(x;\psi)$ is concave in x for each ψ and *continuously generalized differentiable* in x and *continuous differentiable* in ψ on its domain. Finally, we assume that CMESP-UB is a convex program and that its maximum is attained on the feasible set. We let $\mathcal{S}$ denote the feasible set of CMESP-UB, we let $\mathcal{S}^*(\psi) := \{x \in M : f(x;\psi) = z(\psi)\}$ (the optimal x given ψ), and we say that ψ^* is optimal if $z(\psi^*) = \min_{\psi \in T} z(\psi)$.

Remark. *gscaling-linx and gscaling-DDFact can naturally be viewed as an instantiation of CMESP-UB with $\psi := \log \Upsilon$. For gscaling-BQP, we can view $X \in \mathbb{S}^n$ as a vector in $\mathbb{R}^{n(n+1)/2}$, therefore it can also be regarded as an instantiation of CMESP-UB with $\psi := \log \Upsilon$. The continuous generalized differentiability and continuous differentiability of the objective functions is established in the proofs of Theorem 4.1, 4.2, and 4.4.*

We first focus on the cases where $f(x;\psi)$ is convex in ψ, which encompasses gscaling-BQP and gscaling-linx as particular cases. For such cases, $z(\psi)$ becomes a convex function in ψ. Our algorithm relies on the following theorem, tailored from (Zalinescu, 2002, Theorem 2.4.18) to our specific context.

Theorem 4.10. *(Zalinescu, 2002, Theorem 2.4.18) Assume that $f(x;\psi)$ is convex in ψ for every $x \in M$, then the subdifferential of $z(\psi)$ at $\psi \in T$ is*

$$\partial z(\psi) = \overline{conv} \left\{ \frac{f(x;\psi)}{\partial \psi} : x \in \mathcal{S}^*(\psi) \right\},$$

where $\overline{conv}$ denotes the convex closure. Furthermore, if $\mathcal{S}^*(\psi)$ is a singleton, then the unique subgradient becomes the gradient of $z(\psi)$ at ψ.

Remark. *(Fampa and Lee, 2022, Propositions 3.3.7 and 3.6.9) provide sufficient conditions for $\mathcal{S}^*(\psi)$ to be a singleton for gscaling-BQP and gscaling-linx, respectively.*

Theorem 4.10 allows the calculation of the subgradient (or gradient) of $z(\psi)$ by solving CMESP-UB. Thus, a standard subgradient algorithm can achieve convergence to an optimal ψ^*. However, due to the well-known sluggishness of the subgradient algorithm, we employ a BFGS-type algorithm that utilizes the subgradient (or gradient) to update the Hessian approximation.

For cases where $f(x;\psi)$ is not necessarily convex in ψ, we cannot aim for verified global optimality. Nevertheless, we still use a BFGS-type algorithm, where we use $\frac{\partial f(x;\psi)}{\partial \psi}$ for any $x \in \mathcal{S}^*(\psi)$ to update the Hessian approximation. Under some smoothness assumption,

$\frac{\partial f(x;\psi)}{\partial \psi}$ becomes the differential of $z(\psi)$, and this algorithm will converge to a stable point of $z(\psi)$. The following theorem provides a sufficient condition for the differentiability of $z(\psi)$, tailored from (Oyama and Takenawa, 2018, Proposition 2.1) to our specific context.

Theorem 4.11. *(Oyama and Takenawa, 2018, Proposition 2.1) We define a* selection *to be a function mapping from ψ to x selected from $\mathcal{S}^*(\psi)$, denoted as $x^*(\psi)$. Given $\bar{\psi} \in T$, if there is a selection $x^*(\psi)$ continuous at $\bar{\psi}$, then $z(\psi)$ is differentiable at $\bar{\psi}$ with*

$$\frac{\partial z(\bar{\psi})}{\partial \psi} = \frac{\partial f(x^*(\bar{\psi}); \bar{\psi})}{\partial \psi}.$$

In particular, if $M^*(\bar{\psi})$ is a singleton, the unique selection $x^*(\psi)$ is always continuous at $\bar{\psi}$.

Additionally, BFGS has been shown to possess good convergence properties under non-smooth settings, e.g., locally Lipschitz and directionally differentiable (see (Lewis and Overton, 2013)). The following theorem guarantees this property for $z_{\mathrm{DDFact}}(\Upsilon)$, tailored from (Fiacco and Ishizuka, 1990b, Theorem 4.1) to our specific context.

Theorem 4.12. *For any $\psi \in T$, $z(\psi)$ is locally Lipschitz near ψ and directionlly differentiable at ψ in any feasible direction v with formula*

$$\partial z(\psi; v) = \max_{x \in \mathcal{S}^*(\psi)} \left(\frac{\partial f(x;\alpha)}{\partial \alpha} \right)^{\mathsf{T}} v.$$

Remark. *Theorem 4.10, 4.11, and 4.12 hold based on the continuous differentiability of $f(x;\psi)$ in ψ and the continuity of $f(x;\psi)$ in x for CMESP-UB. The continuously generalized differentiability of $f(x;\psi)$ in x ensures good convergence behavior of algorithms for obtaining $x^* \in \mathcal{S}^*(\psi)$.*

4.6 Experiments

We experimented on benchmark instances of MESP, using three covariance matrices that have been extensively used in the literature, with $n = 63, 90, 124$ (see, e.g., (Ko, Lee, and Queyranne, 1995; Lee, 1998; Anstreicher, Fampa, Lee, and Williams, 1999; Anstreicher, 2018, 2020)). For testing CMESP, we included five side constraints $a_i^{\mathsf{T}} x \leq b_i$, for $i = 1, \ldots, 5$, in MESP. As there is no benchmark data for the side constraints, we have generated them randomly. For each n, the left-hand side of constraint i is given by a uniformly-distributed random vector a_i with integer components between -2 and 2. The right-hand side of the constraints was selected so that, for every s considered in the experiment, the best known solution of the instance of MESP is violated by at least one constraint.

For each n (which refers always to a particular benchmark covariance matrix), we consider different values of s defining a set of test instances of MESP and CMESP. We ran our experiments under Windows, on an Intel Xeon E5-2667 v4 @ 3.20 GHz processor equipped with 8 physical cores (16 virtual cores) and 128 GB of RAM. We implemented our code in `Matlab` using the solvers `SDPT3` v. 4.0 for gscaling-BQP, and `Knitro` v. 12.4 for gscaling-linx and gscaling-DDFact. When instantiating the gscaling-DDFact bound, the selection of F is made as $F := U\Lambda^{1/2}$, where $C = U\Lambda U^{\mathsf{T}}$ represents a spectral decomposition of C omitting eigenvalues of zero, so that $U \in \mathbb{R}^{n\times \text{rank}(C)}$ and diagonal matrix $\Lambda \in \mathbb{R}^{\text{rank}(C)\times\text{rank}(C)}$. This choice gives the number of columns of F equal to the rank of C, so that Remark 4.4 applies.

We optimized scaling vectors Υ using a BFGS algorithm, and o-scaling parameters γ using Newton's method. In all of our experiments we set `Knitro` parameters[3] as follows: `convex` = 1 (true), `gradopt` = 1 (we provided exact gradients), `maxit` = 1000. We set `opttol` = 10^{-10}, aiming to satisfy the KKT optimality conditions to a very tight tolerance. We set `xtol` = 10^{-15} (relative tolerance for lack of progress in the solution point) and `feastol` = 10^{-10} (relative tolerance for the feasibility error), aiming for the best solutions that we could reasonably find. In our first set of experiments, we set the the `Knitro` parameter `algorithm` = 3 to use an active-set method. Besides solving the relaxations to get upper bounds for our test instances of MESP and CMESP, we compute lower bounds with a heuristic of (Lee, 1998, Section 4) and then a local search (see (Ko, Lee, and Queyranne, 1995, Section 4)).

In Figure 4.1, we show the impact of g-scaling on the linx bound for MESP on the three benchmark covariance matrices. For the $n = 63$ matrix, we also show the impact of g-scaling on the gscaling-BQP bound. The gscaling-DDFact and complementary gscaling-DDFact bounds are only considered in the experiments for CMESP, as the g-scaling methodology was only able to improve these bounds when side constraints were added to MESP. The plots on the left in Figure 4.1 present the "integrality gap decrease ratios", given by the difference between the integrality gaps using o-scaling and the integrality gaps using g-scaling, divided by the integrality gaps using o-scaling. The integrality gaps are given by the difference between the upper bounds computed with the relaxations and lower bounds given by heuristic solutions. We see that larger n leads to larger maximum ratios. We also see that the g-scaling methodology is effective in reducing all bounds evaluated, especially the linx bound. Even for the most difficult instances, with intermediate values of s, we have some improvement on the bounds, which can be effective in the branch-and-bound context where the bounds would ultimately be applied. The plots on the right in Figure 4.1 present the integrality gaps, and we see that even when the integrality gaps given by the o-scaling

[3] see https://www.artelys.com/docs/knitro/2_userGuide.html, for details

are less than 1, g-scaling can reduce them.

In Figure 4.2, we show for CMESP, similar results to the ones shown in Figure 4.1, except that now we also present the effect of g-scaling on the gscaling-DDFact and the complementary gscaling-DDFact bounds. We see from the integrality gap decrease ratios that when side constraints are added to MESP, the g-scaling is, in general, more effective in reducing the gaps given by o-scaling. We also see that, it is particularly effective in reducing the gscaling-DDFact and complementary gscaling-DDFact bounds that were not improved by o-scaling. Especially for the $n = 124$ matrix, we see a significant reduction on the gaps given by complementary gscaling-DDFact and gscaling-DDFact, for s smaller and greater than 50, respectively.

We also investigated how the improvement of g-scaling over o-scaling for the linx bound can increase the possibility of fixing variables in MESP and CMESP. The methodology for fixing variables is based on convex duality and has been applied since the first convex relaxation was proposed for these problems in (Anstreicher, Fampa, Lee, and Williams, 1996). When a lower bound for each instance is available, the dual solution of the relaxation can potentially be used to fix variables at 0/1 values (see (Fampa and Lee, 2022), for example). This is an important feature in the B&B context. The methodology may be able to fix a number of variables when the relaxation generates a strong bound, and in doing so, it reduces the size of the successive subproblems and improves the bounds computed for them.

In Table 4.1, for MESP, we consider (un-scaled) gscaling-DDFact and (un-scaled) complementary gscaling-DDFact, and we show the impact of using g-scaled gscaling-linx, compared to o-scaled gscaling-linx, on an iterative procedure where we solve gscaling-linx, gscaling-DDFact, and complementary gscaling-DDFact, fixing variables at 0/1 whenever possible. While for CMESP, we show the impact of using g-scaled linx, g-scaled gscaling-DDFact, and g-scaled complementary gscaling-DDFact, compared to o-scaled linx, (un-scaled) gscaling-DDFact, and (un-scaled) complementary gscaling-DDFact, on the same iterative procedure where we solve gscaling-linx, gscaling-DDFact, and complementary gscaling-DDFact, fixing variables at 0/1 whenever possible. In both cases, we update the scaling parameters of the scaled bounds at every iteration. For o-scaling, we optimize the scalar γ by applying Newton steps until the absolute value of the derivative is less than 10^{-10}. For g-scaling, we optimize the vector Υ by applying up to 10 BFGS steps, taking $\gamma\mathbf{e}$ as a starting point. We limit the number of BFGS steps in this experiment to get closer to what might be practical within B&B. We present in the columns of Table 4.1, the following information from left to right: The problem considered, n, the range of s considered, the scaling, the number of instances solved (one for each s considered), the number of instances on which we could fix at least one variable ("inst fix"), the total number of variables fixed on all instances solved ("var fix"),

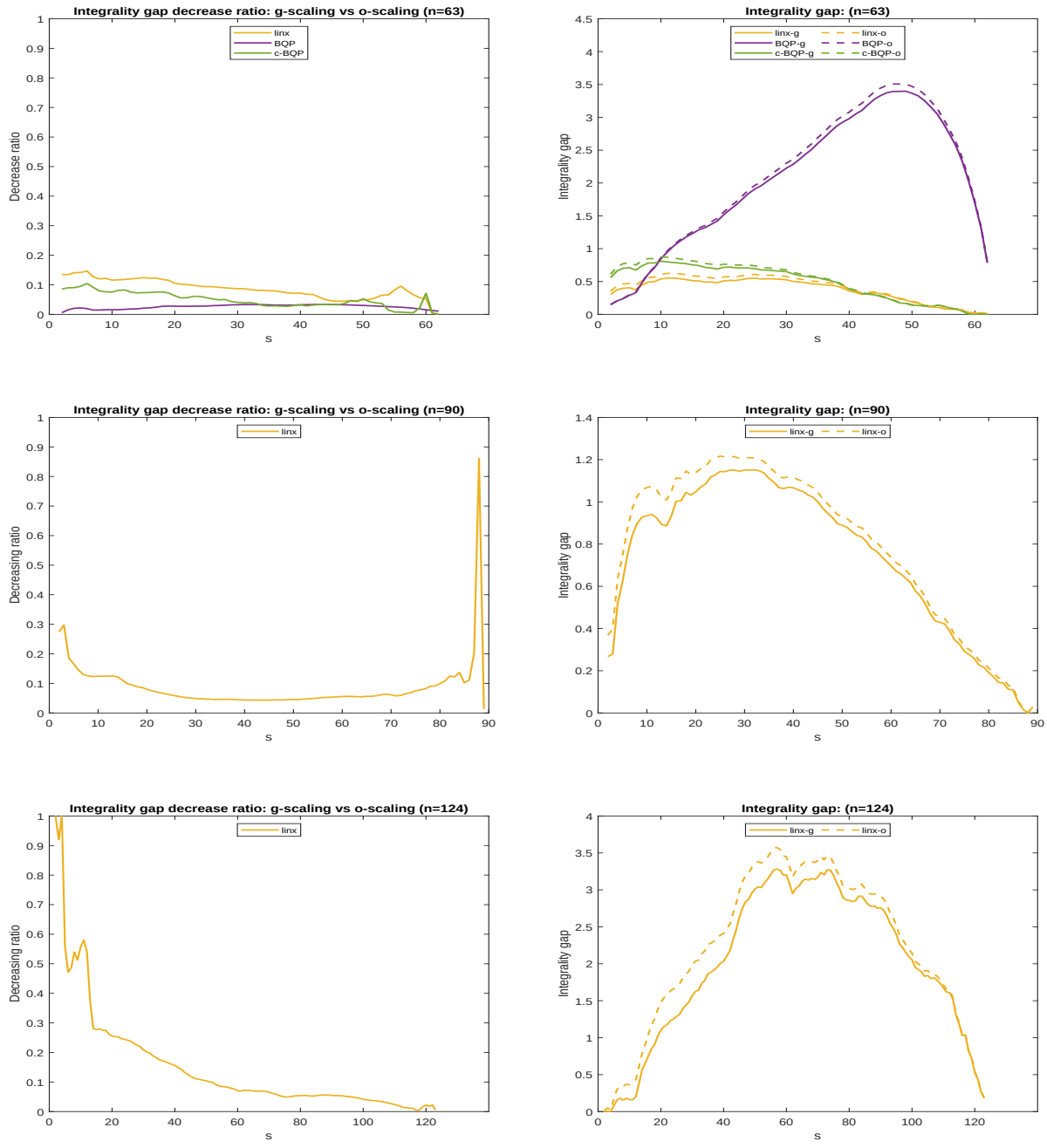

Figure 4.1: Comparison between g-scaling and o-scaling for MESP

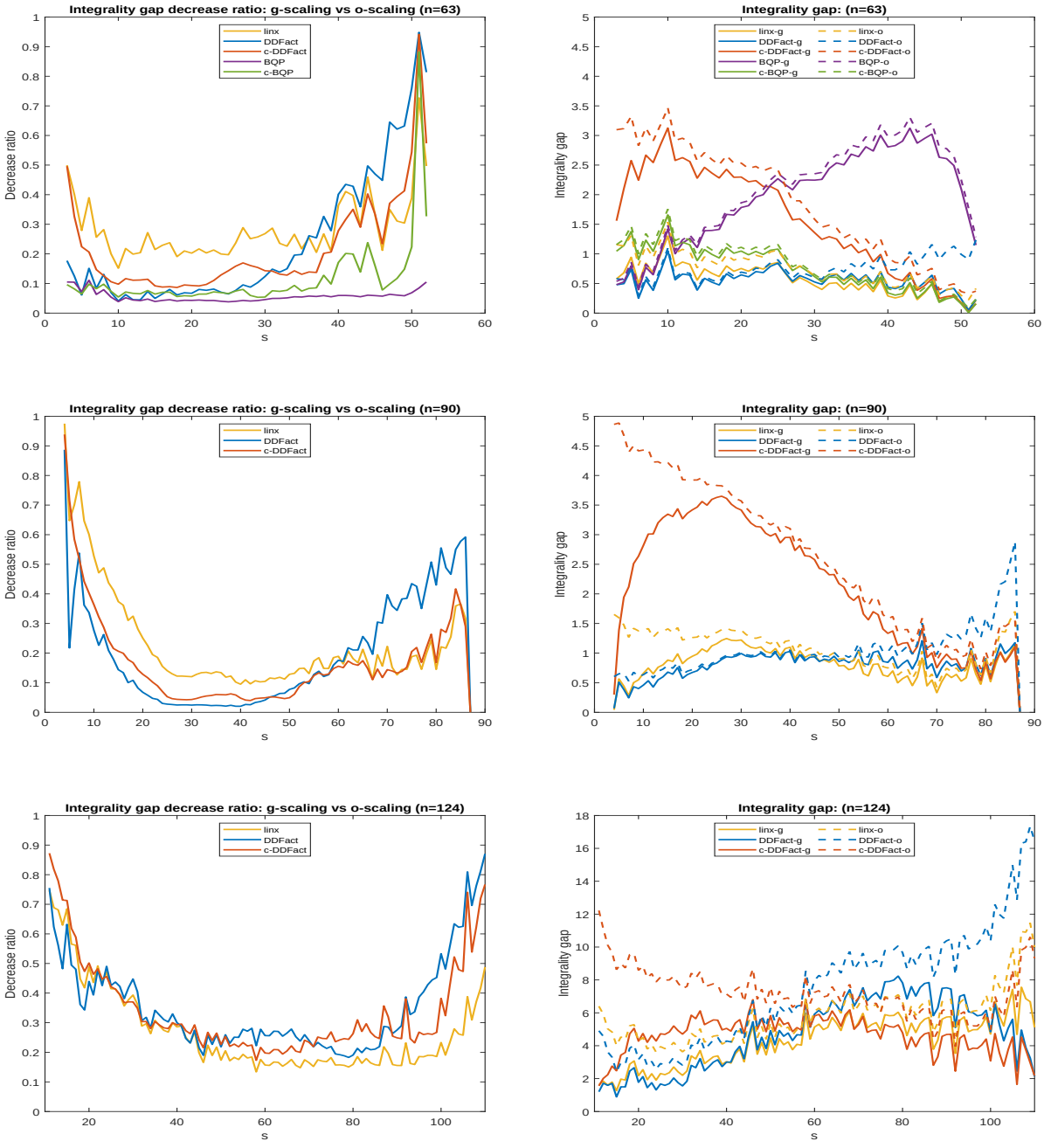

Figure 4.2: Comparison between g-scaling and o-scaling for CMESP

the %-improvement of g-scaling over o-scaling for the two last statistics. Additionally, to better understand how well our methods works for MESP as n grows, we also experimented with a covariance matrix of order $n = 300$, which is a principal submatrix of the covariance matrix of order $n = 2000$ used as a benchmark in the literature (see (Li and Xie, 2023; Chen, Fampa, and Lee, 2023)). First, we see that, except for the number of instances of MESP with $n = 124$ and $n = 300$ on which we could fix variables, there is always an improvement. The improvement becomes very significant when side constraints are considered. We note that the number of variables fixed, reported on Table 4.1, refers only to the root nodes of the B&B algorithm and indicates a promising approach to reduce the B&B enumeration.

				Number of			Improvement	
					inst	var	inst	var
	n	s	scal	s	fix	fix	fix	fix
MESP	63	[2,62]	o	61	41	1123		
			g	61	42	1140	2.44%	1.51%
	90	[2,89]	o	88	41	1741		
			g	88	42	1790	2.44%	2.81%
	124	[2,123]	o	122	35	3322		
			g	122	35	3353	0.00%	0.93%
	300	[80,120]	o	41	41	8382		
			g	41	41	10753	0.00%	28.3%
CMESP	63	[3,52]	o	50	22	371		
			g	50	28	537	27.27%	44.74%
	90	[4,87]	o	84	26	606		
			g	84	37	1048	42.31%	72.94%
	124	[11,110]	o	100	9	197		
			g	100	33	1120	266.67%	468.53%

Table 4.1: Impact of g-scaling on variable fixing

The experiments with the fixing methodology show that g-scaling can effectively lead to a positive impact on the solution of MESP and CMESP, especially of the latter.

We carried out further experiments to investigate the relevance of our generalized differentiability for gscaling-DDFact. For these experiments, we only worked with MESP, because we wanted to better expose the non-negativity constraints to the algorithms, and we chose (again) factorizations with k equal to the rank of C, taking advantage of Remark 4.4. For these experiments, we employed all four of the `Knitro` algorithmic options: Interior/Direct, Interior/Conjugate-Gradient(CG), Active Set, Sequential Quadratic Programming (SQP), and chose all of the other `Knitro` parameters as described for our first experiments. The first two algorithms have all of their iterates in the interior of $\mathrm{dom}\,(f_{\mathrm{DDFact}}; \Upsilon)_+$, while the

n	Interior	Interior(CG)	Active-set	SQP
63	0.09	0.42	0.11	0.18
90	0.19	0.83	0.20	0.29
124	0.40	1.63	0.37	0.44
2000	1292.2	2227.39	96.30	304.60

Table 4.2: Average converging time of each algorithm for solving gscaling-DDFact.

latter two can have iterates at the boundary of $\mathrm{dom}\,(f_{\text{DDFact}};\Upsilon)_+$. We collected average converging times of the four algorithms in Table 4.2. In particular, the converging times are averaged over $5 \le s \le n-5$ for the $n = 63, 90, 124$ benchmark covariance matrices and over $s = 20, 40, 60, 80, 100$ for the (full) $n = 2000$ benchmark covariance matrix. To mitigate the impact of some variance in the run time for each instance, we also included in Table 4.3 the percentage of instances s for each n where the convergence time of the algorithm is within 105% of the convergence time of the best-performing algorithm among the four. This criterion implies that the algorithm is considered the best within a tolerance of 5%. Additionally, in Table 4.4, we gathered the average iterates that lie on the boundary of $\mathrm{dom}\,(f_{\text{DDFact}};\Upsilon)_+$ for each algorithm to exhibit the relevance of generalized differentiabiliy in Definition 4.2. We use the rank function of `MATLAB` to determine the boundary iterates by singularity of $F_{\text{DDFact}}(x;\Upsilon)$ (equivalently, when x has any zero components; see Remark 4.4). In particular, `MATLAB` asserts a matrix to be singular if the matrix has some singular value smaller than the product of the maximum of dimension lengths and the exponential of the matrix 2-norm.

Table 4.2 exhibits that the active-set algorithm consistently achieves the minimum, or near-minimum, average converging time. Table 4.3 shows that the active-set algorithm has the greatest winning percentages except for $n = 124$, where the combined winning percentages of the active-set and SQP algorithms still exceed those of the other two algorithms. These outcomes indicate the superiority of algorithms that produce iterates lying on the boundary of $\mathrm{dom}\,(f_{\text{DDFact}};\Upsilon)_+$, thereby emphasizing the relevance of generalized differentiability in justifying their use. Table 4.4 reveals that for both the active-set and SQP algorithms, nearly all iterates are on the boundary of $\mathrm{dom}\,(f_{\text{DDFact}};\Upsilon)_+$. We note that even the interior-point methods display iterates on the boundary of $\mathrm{dom}\,(f_{\text{DDFact}};\Upsilon)_+$ within the tolerance. These findings underscore the relevance of generalized differentiability across all algorithms.

n	Interior	Interior(CG)	Active-set	SQP
63	46.3	0	55.6	0
90	48.8	0	57.3	1.2
124	48.3	0	44.8	13.8
2000	0	0	100.0	0

Table 4.3: % of s on which the algorithm converges within no more than 105% converging time of the best algorithm (i.e., optimal under 5% tolerance).

n	Interior	Interior(CG)	Active-set	SQP
63	27.6	1.3	97.7	97.2
90	35.5	0.8	98.7	98.6
124	61.1	22.4	93.0	93.6
2000	69.2	29.5	100.0	100.0

Table 4.4: Average % of iterates with x having any zero components, which is equivalent to the singularity of $F_{\text{DDFact}}(x; \Upsilon)$.

4.7 Concluding remarks

We have seen that g-scaling can lead to improvements in upper bounds and variable fixing for MESP and very good improvements for CMESP. In future work, we will implement this in an efficient manner, within a B&B algorithm. In that context, it is important to efficiently use parent scaling vectors to warm-start the optimization of scaling vectors for children (see (Anstreicher, 2020), where this was an important issue for o-scaling in the context of the gscaling-linx bound). An open question that we wish to highlight is whether g-scaling can help the gscaling-DDFact bound for MESP. Theorem 4.4.*iv* is a partial result toward a negative answer.

Finally, we remark that there is room to do g-scaling for other bounds for CMESP. We did not work with g-scaling for the NLP bound. Besides the fact that we do not have a convexity result for o-scaling of the NLP bound as a starting point for generalizing the theory, the o-scaling parameter is entangled with other parameters of the NLP bound which must be selected properly (even for the NLP bound to be a convex optimization problem). For these reasons, we have left exploration of g-scaling for the NLP bound for future research. Additionally, we did not attempt to merge the ideas of g-scaling with bound "mixing" (see (Chen, Fampa, Lambert, and Lee, 2021)); this looks like another promising area for investigation.

CHAPTER 5

Masking Anstreicher's linx Bound for Improved Entropy Bounds

This chapter has been published as:

Zhongzhu Chen, Marcia Fampa, Jon Lee. Technical Note: Masking Anstreicher's linx bound for improved entropy bounds. *Operations Research*, appeared online, 2022. https://doi.org/10.1287/opre.2022.2324

5.1 Introduction

The main goal of this chapter is to demonstrate the strong potential for masking to improve the linx bound, even in the presence of scaling. In fact, we show that for a large class of problem instances, masking can improve the linx bound by an amount that is at least linear in the problem size n. In detail, we exhibit sequences $\{C_k, s_k; M_k, \gamma_k, \hat{\gamma}_k\}_{k=1}^{\infty}$, with $C_k \succeq 0$ of order n_k, $M_k \in \mathcal{M}_{n_k}$, $n_k \to \infty$, such that $z_{\text{linx}}(C_k, s_k; \gamma_k) - z_{\text{linx}}(C_k, s_k; M_k, \hat{\gamma}_k) \geq \alpha_k$, where α_k grows linearly with n_k. First we do this for $\gamma_k = \hat{\gamma}_k = 1$ (i.e., no scaling). Then, at the expense of a worse lower bound α_k, we do this when γ_k and $\hat{\gamma}_k$ are the optimal scale factors. To get such lower bounds on the gap $z_{\text{linx}}(C_k, s_k; \gamma_k) - z_{\text{linx}}(C_k, s_k; M_{n_k}, \hat{\gamma}_k)$, we need a good lower bound on $z_{\text{linx}}(C_k, s_k; \gamma_k)$ and a good upper bound on $z_{\text{linx}}(C_k, s_k; M_{n_k}, \hat{\gamma}_k)$. In fact, to establish these gaps, we will take $M_{n_k} = I_{n_k}$, and so to get our needed upper bounds, we use an exact characterization of the linx bound and the optimal scaling for the linx bound, when C is diagonal (useful because $C_{n_k} \circ I_{n_k}$ is diagonal).

Because we are going to use masking and scaling simultaneously, we introduce the *masked scaled linx* bound here. For $x \in [0,1]^n$, we define

$$f_{\text{linx}}(x; M, \gamma) := \tfrac{1}{2}\left(\text{ldet}\left(\gamma(C \circ M)\,\text{Diag}(x)(C \circ M) + \text{Diag}(\mathbf{e} - x)\right) - s\log\gamma\right)$$

with

$$\operatorname{dom}(f_{\text{linx}}; M, \gamma) := \left\{ x \in \mathbb{R}^n : \gamma(C \circ M) \operatorname{Diag}(x)(C \circ M) + \operatorname{Diag}(\mathbf{e} - x) \succ 0 \right\}.$$

We then define the *masked scaled linx* bound

$$z_{\text{linx}}(M, \gamma) := \max \left\{ f_{\text{linx}}(x; M, \gamma) \ : \ x \in P_{\text{linx}}(n, s) \right\}. \qquad \text{(masked scaled linx)}$$

where $P_{\text{linx}}(n, s) := \{\mathbf{e}^{\mathsf{T}} x = s,\ 0 \leq x \leq \mathbf{e},\ Ax \leq b\}$.

Moreover, we also extend an earlier result that the scaled linx bound is convex in the logarithm of a scaling parameter, making a full characterization of its behavior and providing an efficient means of calculating its limiting behavior in all cases. Therefore, we introduce a basic results here for following use. Because the scaled linx is an "exact relaxation" (i.e., the objective of the relaxation on an $x \in \{0,1\}^n$ is exactly $\operatorname{ldet} C[S,S]$ for S equal to the support of x), for every scaling parameter $\gamma > 0$, the following useful fact (true for any exact relaxation) is easy to see.

Proposition 5.1. *If $\hat{x} \in \{0,1\}^n$ is an optimal solution of the scaled linx for $\gamma = \hat{\gamma}$, then $\hat{\gamma}$ is optimal. That is, $z_{linx}(C, s; \hat{\gamma}) = \min_{\gamma > 0} z_{linx}(C, s; \gamma)$.*

We note that masking does not generally produce an exact relaxation, so Proposition 5.1 does not extend to a sufficient condition for optimal masks.

In §5.2, we establish that using a mask but no scaling parameter (i.e., $\gamma = 1$), the best-case improvement in the linx bound is at least linear in n; specifically, $\approx .0312n$. In §5.3, we study the behavior of the linx bound as we vary the scaling parameter $\gamma > 0$. It was already established that the linx bound is convex in $\log(\gamma)$ (see (Chen, Fampa, Lambert, and Lee, 2021)). We establish the limiting behavior, as γ goes to 0 and to infinity. When $s = \operatorname{rank}(C)$, the limit as γ goes to infinity can be better than any finite choice of γ; in this case, we establish that the limit can be calculated by solving a single convex optimization problem. In §5.4, we establish that using a mask and *optimal* scaling parameters, the best-case improvement in the linx bound remains at least linear in n; specifically, $\approx .024n$. §5.5 contains some final remarks.

5.2 Linear gap for the linx bound

Our main goal in this section is to establish a linear lower bound on the best-case gap between the linx bound and the masked linx bound, giving a good justification for considering mask optimization. Specifically, we will give a sequence $\{C_n, s_n; M_n\}$, for all even positive integers

n, with $C_n \succeq 0$ of order n, and $M_n \in \mathcal{M}_n$, such that $z_{\text{linx}}(C_n, s_n) - z_{\text{linx}}(C_n, s_n; M_n) \geq \frac{1}{4}\log\left(\frac{4}{3}\right) n$. In fact, we will take $s_n := \frac{n}{2}$, and $M_n := I_n$. Because we use $M_n := I_n$, we will have $z_{\text{linx}}(C_n, s_n; M_n) = z_{\text{linx}}(\text{Diag}(d_{(n)}), s_n)$, where $d_{(n)} = \text{diag}(C_n)$. Hence it is useful to characterize, in general, the optimal solution of linx when C is diagonal. Additionally, beyond our own use in the present work, we believe that such a characterization can be useful in future work on gaps for the linx bound.

Without loss of generality, we assume that $C := \text{Diag}(d)$ where $d \in \mathbb{R}^n$ and $d_1 \geq d_2 \geq \cdots \geq d_n > 0$. Then

$$f_{\text{linx}}(C, s; x) = \tfrac{1}{2}\log \prod_{i \in N} \left((d_i^2 - 1)x_i + 1\right).$$

Lemma 5.2. *Let $C := \text{Diag}(d)$, where $d \in \mathbb{R}^n$ satisfies $d_1 \geq d_2 \geq \cdots \geq d_n > 0$. There exists an optimal solution $\hat{x}$ of linx such that $\hat{x}_1 \geq \hat{x}_2 \geq \cdots \geq \hat{x}_n$ and $\hat{x}_i = \hat{x}_j$, for all $i, j \in N$, such that $d_i = d_j$.*

Proof. Clearly, linx has an optimal solution $\hat{x}$. And

$$\begin{aligned}
&\left((d_i^2 - 1)\hat{x}_i + 1\right)\left((d_j^2 - 1)\hat{x}_j + 1\right) \\
&-\left((d_i^2 - 1)\hat{x}_j + 1\right)\left((d_j^2 - 1)\hat{x}_i + 1\right) \\
&= (d_i^2 - d_j^2)(\hat{x}_i - \hat{x}_j).
\end{aligned}$$

If $d_i > d_j$, from the identity above we see that $\hat{x}_i \geq \hat{x}_j$, otherwise, by exchanging components i and j of $\hat{x}$, we would increase the objective value of linx.

If $d_i = d_j$, let $\delta := \hat{x}_i + \hat{x}_j$. Then,

$$\begin{aligned}
&\left((d_i^2 - 1)\hat{x}_i + 1\right)\left((d_j^2 - 1)\hat{x}_j + 1\right) \\
&\quad = \left((d_i^2 - 1)\hat{x}_i + 1\right)\left((d_j^2 - 1)(\delta - \hat{x}_i) + 1\right).
\end{aligned}$$

In this case, if $d_i = 1$, the above function is constant. Otherwise, it is a univariate concave quadratic in $\hat{x}_i$, and its maximum is uniquely attained at $\hat{x}_i = \delta/2$. Therefore, by setting $\hat{x}_i = \hat{x}_j = \delta/2$, the maximum of the function is always attained. □

Definition 5.1. *We refer to an optimal solution $\hat{x}$ of linx which satisfies the properties in Lemma 5.2 as a uniform optimal solution.*

Lemma 5.3. *Let $C := \text{Diag}(d)$, where $d \in \mathbb{R}^n$ satisfies $d_1 \geq d_2 \geq \cdots \geq d_n > 0$ and $0 \leq x \leq \mathbf{e}$. Then $f_{linx}(C, s; x)$ strictly increases with x_i, if $d_i > 1$, does not change with x_i, if $d_i = 1$, and strictly decreases with x_i, if $d_i < 1$. Furthermore, $f_{linx}(C, s; x)$ is concave in $[0, 1]^n$ and strictly concave if $d_i \neq 1$, for all $i \in N$.*

Proof. For all $i \in N$, $d_i > 0$ and $0 \le x_i \le 1$ implies that $(d_i^2 - 1)x_i + 1 > 0$. Then, for $i \in N$,

$$\frac{\partial f_{\text{linx}}(C, s; x)}{\partial x_i} = \frac{d_i^2 - 1}{2((d_i^2 - 1)x_i + 1)} \begin{cases} < 0, \text{ if } d_i < 1, \\ = 0, \text{ if } d_i = 1, \\ > 0, \text{ if } d_i > 1, \end{cases}$$

$$\frac{\partial^2 f_{\text{linx}}(C, s; x)}{\partial x_i^2} = \frac{-(d_i^2 - 1)^2}{2((d_i^2 - 1)x_i + 1)^2} \begin{cases} < 0, \text{ if } d_i \neq 1, \\ = 0, \text{ if } d_i = 1, \end{cases}$$

and

$$\frac{\partial^2 f_{\text{linx}}(C, s; x)}{\partial x_i \partial x_j} = 0, \text{ for } 1 \le i \neq j \le n.$$

□

Remark. *From Lemmas 5.2 and 5.3, we see that if $C = \text{Diag}(d)$, with $0 < d_i \neq 1, \forall i \in N$, then linx has a unique optimal solution, which is a uniform optimal solution.*

Next, we establish necessary conditions for $\hat{x}$ to be a uniform optimal solution for linx when C is diagonal, based on checking a finite set of feasible directions; we could also get these conditions from the KKT conditions for linx, also establishing their sufficiency, but our approach is simpler and suits our purpose.

Lemma 5.4. *Let $C := \text{Diag}(d)$, where $d \in \mathbb{R}^n$ satisfies $d_1 \ge d_2 \ge \cdots \ge d_n > 0$. Let $\hat{x}$ be a uniform optimal solution of linx. For $1 \le i < j \le n$, we have*

$$\frac{d_j^2 - 1}{(d_j^2 - 1)\hat{x}_j + 1} \le \frac{d_i^2 - 1}{(d_i^2 - 1)\hat{x}_i + 1}. \tag{5.1}$$

Additionally, if $1 > \hat{x}_i \ge \hat{x}_j > 0$, then

$$\frac{d_j^2 - 1}{(d_j^2 - 1)\hat{x}_j + 1} = \frac{d_i^2 - 1}{(d_i^2 - 1)\hat{x}_i + 1}. \tag{5.2}$$

Proof. (5.1) is clear when $\hat{x}_i = \hat{x}_j$, from the fact that $d_i \ge d_j > 0$. So we may assume that $\hat{x}_i > \hat{x}_j$. In this case $\mathbf{e}_j - \mathbf{e}_i$ is a feasible direction for $\hat{x}$ relative to linx. Because $\hat{x}$ is optimal for linx, we must have that $\nabla f_{\text{linx}}(C, s; \hat{x})^{\mathsf{T}}(\mathbf{e}_j - \mathbf{e}_i) \le 0$, which is equivalent to (5.1).

(5.2) follows from the fact that, in this case, $\mathbf{e}_i - \mathbf{e}_j$ is also a feasible direction for $\hat{x}$ relative to linx. □

We have a corollary of Lemma 5.4 for two special cases: $d_n > 1$ and $d_1 < 1$. We will see later that the characterization of the optimal solution in general can be reduced to the characterization of the optimal solution in these two special cases.

Corollary 5.5. *Let $C := \mathrm{Diag}(d)$, where $d \in \mathbb{R}^n$ satisfies either $d_1 \geq d_2 \geq \cdots \geq d_n > 1$ or $1 > d_1 \geq d_2 \geq \cdots \geq d_n > 0$. Let $\hat{x}$ be a uniform optimal solution of linx. Then,*

$$\hat{x}_i - \hat{x}_j \leq \frac{1}{d_j^2 - 1} - \frac{1}{d_i^2 - 1}. \tag{5.3}$$

Additionally, if $1 > \hat{x}_i \geq \hat{x}_j > 0$, then

$$\hat{x}_i - \hat{x}_j = \frac{1}{d_j^2 - 1} - \frac{1}{d_i^2 - 1}. \tag{5.4}$$

Proof. If either $d_n > 1$ or $d_1 < 1$, we have $d_i^2 - 1 \neq 0, \forall i \in N$. Also, both $d_i^2 - 1$ and $d_j^2 - 1$ have the the same sign $\forall i, j \in N$. Together with $(d_i^2 - 1)\hat{x}_i + 1 > 0, \forall i \in N$, we have that (5.1) and (5.2) equal (5.3) and (5.4), respectively. □

To characterize an optimal solution of linx when C is diagonal, we first establish a lemma that characterizes an optimal solution of linx in the two special cases discussed in Corollary 5.5.

Lemma 5.6. *Let $C := \mathrm{Diag}(d)$, where $d \in \mathbb{R}^n$ satisfies either $d_1 \geq d_2 \geq \cdots \geq d_n > 1$ or $1 > d_1 \geq d_2 \geq \cdots \geq d_n > 0$. Let $\hat{x}$ be a uniform optimal solution of linx for a given $0 < s < n$. We have,*

(i) if $\frac{1}{d_{s+1}^2 - 1} - \frac{1}{d_s^2 - 1} \geq 1$, then

$$\hat{x}_i := \begin{cases} 1, & \text{for } 1 \leq i \leq s, \\ 0, & \text{for } s+1 \leq i \leq n, \end{cases}$$

(ii) if $\frac{1}{d_{s+1}^2 - 1} - \frac{1}{d_s^2 - 1} < 1$, then $0 < \hat{x}_s < 1$, and

$$\hat{x}_i := \begin{cases} \min\left\{1,\ \hat{x}_s + \frac{1}{d_s^2 - 1} - \frac{1}{d_i^2 - 1}\right\}, \\ \qquad\qquad \text{for } 1 \leq i \leq s-1, \\ \max\left\{0,\ \hat{x}_s + \frac{1}{d_s^2 - 1} - \frac{1}{d_i^2 - 1}\right\}, \\ \qquad\qquad \text{for } s+1 \leq i \leq n. \end{cases} \tag{5.5}$$

Proof. We have already shown that under the hypotheses, linx has a unique optimal solution $\hat{x}$, where $\hat{x}_1 \geq \hat{x}_2 \geq \cdots \geq \hat{x}_n$. Thus, for (i), we only need to show that $\hat{x}_s = 1$. Suppose that $\hat{x}_s < 1$; then $1 > \hat{x}_s \geq \hat{x}_{s+1} > 0$, i.e., $\hat{x}_s - \hat{x}_{s+1} < 1 \leq \frac{1}{d_{s+1}^2 - 1} - \frac{1}{d_s^2 - 1}$, which violates the necessary condition (5.4) in Corollary 5.5.

For (ii), we see that if $\hat{x}_s = 1$, then $\hat{x}_{s+1} = 0$ and $\hat{x}_s - \hat{x}_{s+1} = 1 > \frac{1}{d_{s+1}^2-1} - \frac{1}{d_s^2-1}$, which violates the necessary condition (5.3) in Corollary 5.5. If $\hat{x}_s = 0$, then $\sum_{i=1}^{n} \hat{x}_i \leq \sum_{i=1}^{s-1} \hat{x}_i \leq s-1$, which contradicts the feasibility of $\hat{x}$. Therefore, $0 < \hat{x}_s < 1$. Finally, by the necessary conditions in Corollary 5.5, the other parts of (ii) must hold. □

In case (ii) in Lemma 5.6, we can solve the equation $\mathbf{e}^{\mathsf{T}} x = s$ for $\hat{x}_s$:

$$\sum_{i=1}^{s-1} \min\left\{1,\ \hat{x}_s + \frac{1}{d_s^2-1} - \frac{1}{d_i^2-1}\right\} + \hat{x}_s + \sum_{i=s+1}^{n} \max\left\{0,\ \hat{x}_s + \frac{1}{d_s^2-1} - \frac{1}{d_i^2-1}\right\} = s\ ,$$

where $0 < \hat{x}_s < 1$. Note that the left-hand side of this equation is increasing, piecewise linear, and continuous in $\hat{x}_s$, so the equation is easy to solve. Once $\hat{x}_s$ is determined, all $\hat{x}_i$, with $i \neq s$ are also uniquely determined by (5.5).

Finally, we have the following characterization of optimal solutions when C is diagonal. Below, we use $\mathcal{L}(\tilde{C}, \tilde{s})$ to denote linx with (C, s) replaced by $(\tilde{C}, \tilde{s})$.

Theorem 5.7. *Let $C := \mathrm{Diag}(d)$, where $d \in \mathbb{R}^n$ satisfies $d_1 \geq d_2 \geq \cdots \geq d_n > 0$. Let $L := \{i \in N : d_i < 1\}$, $E := \{i \in N : d_i = 1\}$, and $G := \{i \in N : d_i > 1\}$. Then $\hat{x}$ defined below is an optimal solution for linx.*

(i) If $s \leq |G|$, let $\tilde{x} \in \mathbb{R}^{|G|}$ be the optimal solution of $\mathcal{L}(\tilde{C}, \tilde{s})$ with $\tilde{C} := \mathrm{Diag}(\tilde{d})$, where $\tilde{d} \in \mathbb{R}^{|G|}, \tilde{d}_i := d_i, 1 \leq i \leq |G|$, and $\tilde{s} := s$. Then,

$$\hat{x}_i := \begin{cases} \tilde{x}_i, & \text{for } i \in G, \\ 0, & \text{for } i \in E \cup L. \end{cases}$$

(ii) If $|G| < s \leq |G \cup E|$, let $\tilde{x}_i, i \in E$ be any value such that $0 \leq \tilde{x}_i \leq 1$ and $\sum_{i \in E} \tilde{x}_i = s - |G|$. Then,

$$\hat{x}_i := \begin{cases} 1, & \text{for } i \in G, \\ \tilde{x}_i, & \text{for } i \in E, \\ 0, & \text{for } i \in L. \end{cases}$$

(iii) If $|G \cup E| < s$, let $\tilde{x} \in \mathbb{R}^{|L|}$ be the optimal solution of $\mathcal{L}(\tilde{C}, \tilde{s})$ with $\tilde{C} := \mathrm{Diag}(\tilde{d})$ where $\tilde{d} \in \mathbb{R}^{|L|}, \tilde{d}_i = d_{i+|G \cup E|}, 1 \leq i \leq |L|$, and $\tilde{s} := s - |G \cup E|$. Then,

$$\hat{x}_i := \begin{cases} 1, & \text{for } i \in G \cup E, \\ \tilde{x}_i, & \text{for } i \in L. \end{cases}$$

Proof. We will prove (i) in detail; (ii) and (iii) can be proved in a similar manner. The feasibility of $\hat{x}$ is obvious. We will prove that $\hat{x}$ is optimal as well. Let us assume otherwise, i.e., we assume that x^* is an optimal solution to linx and

$$f_{\text{linx}}(C, s; x^*) > f_{\text{linx}}(C, s; \hat{x}). \tag{5.6}$$

We first claim that $x^*_i = 0$, $\forall i \in E \cup L$. Otherwise let $x^*_i > 0$, for some $i \in E \cup L$. Then, as $s \leq |G|$, by feasibility of x^*, we have $x^*_j < 1$ for some $j \in G$. Therefore $e_j - e_i$ is a feasible direction from x^* in linx. However, by Lemma 5.3, we have that $\nabla f_{\text{linx}}(C, s; x^*)^{\mathsf{T}}(e_j - e_i) > 0$, contradicting the optimality of x^*.

Now, define $\tilde{x}^* \in \mathbb{R}^{|G|}$ such that $\tilde{x}^*_i = x^*_i$, $\forall i \in G$. As $x^*_i = 0$, $\forall i \in E \cup L$, it is straightforward to see that $\tilde{x}^*$ is feasible to $\mathcal{L}(\tilde{C}, \tilde{s})$. Thus $f_{\text{linx}}(\tilde{C}, \tilde{s}; \tilde{x}) \geq f_{\text{linx}}(\tilde{C}, \tilde{s}; \tilde{x}^*)$. Note also that $\hat{x}_i = 0$, $\forall i \in E \cup L$, so $f_{\text{linx}}(C, s; \hat{x}) = f_{\text{linx}}(\tilde{C}, \tilde{s}; \tilde{x}) \geq f_{\text{linx}}(\tilde{C}, \tilde{s}; \tilde{x}^*) = f_{\text{linx}}(C, s; x^*)$, contradicting (5.6).

□

Having characterized an optimal solution for linx when C is diagonal, we will now consider the more general case where C is any positive-semidefinite matrix and establish a simple lower bound on $z_{\text{linx}}(C, s)$, by considering the eigenvalues of C.

Lemma 5.8. *For any positive-semidefinite order-n matrix C and integer $0 < s < n$,*

$$z_{linx}(C, s) \geq \tfrac{1}{2} \log \prod_{i=1}^{n} \left(\tfrac{s}{n} \lambda_i^2 + 1 - \tfrac{s}{n} \right),$$

where $\lambda_1, ..., \lambda_n$ are the eigenvalues of C.

Proof. We diagonalize C: That is, we choose an orthogonal matrix Q so that $Q^{\mathsf{T}} C Q = \Lambda := \text{Diag}(\lambda_1, \lambda_2, ..., \lambda_n)$. Let $\bar{x} = \frac{s}{n}\mathbf{e}$. Then

$$\begin{aligned}
z_{\text{linx}}(C, s) \geq & f_{\text{linx}}(C, s; \bar{x}) \\
= & \tfrac{1}{2} \operatorname{ldet} \left(\tfrac{s}{n} C^2 + \left(1 - \tfrac{s}{n}\right) I \right) \\
= & \tfrac{1}{2} \operatorname{ldet} \left(\tfrac{s}{n} Q \Lambda^2 Q^{\mathsf{T}} + \left(1 - \tfrac{s}{n}\right) I \right) \\
= & \tfrac{1}{2} \operatorname{ldet} \left(\tfrac{s}{n} \Lambda^2 + \left(1 - \tfrac{s}{n}\right) I \right) \\
= & \tfrac{1}{2} \log \prod_{i=1}^{n} \left(\tfrac{s}{n} \lambda_i^2 + 1 - \tfrac{s}{n} \right).
\end{aligned}$$

□

We now study the efficacy of using a mask M for the linx bound (versus choosing $M = J$). We will show that there is an infinite sequence of $\{C_n\}_{n\in\mathcal{I}}$ such that $z_{\text{linx}}(C_n, \frac{n}{2}) - z_{\text{linx}}(C_n, \frac{n}{2}; I) \geq \frac{1}{4}\log(\frac{4}{3})n \approx .0312n$. The results show that by choosing an appropriate mask M different from J, we can decrease the linx bound by at least an amount that is linear in n.

Recall that we have characterized an optimal solution of linx when C is diagonal in Theorem 5.7 and a lower bound of $z_{\text{linx}}(C, s)$ when C is any positive-semidefinite matrix in Lemma 5.8. Note that $C \circ I$ is diagonal. Then we have the following gap.

$$\begin{aligned} &z_{\text{linx}}(C, s) - z_{\text{linx}}(C, s; I) \geq \\ &\tfrac{1}{2}\log\prod_{i=1}^{n}\left(\tfrac{s}{n}\lambda_i^2 + 1 - \tfrac{s}{n}\right) - \tfrac{1}{2}\log\prod_{i=1}^{n}\left(d_i^2\hat{x}_i + 1 - \hat{x}_i\right) \end{aligned} \tag{5.7}$$

where $\lambda_i, i \in N$ are the eigenvalues of C, $d_i, i \in N$ are diagonal elements of C, and $\hat{x}$ is an optimal solution of linx with C replaced by $C \circ I$. We will employ this lower bound on the gap in what follows.

Before presenting our main result, we will characterize the optimal mask when $n = 2$ and $s = 1$. We will use this to construct a gap between $z_{\text{linx}}(C, s)$ and $z_{\text{linx}}(C, s; I)$ that is linear in the order of C.

Theorem 5.9. *Let* $C_2 := \begin{pmatrix} a & c \\ c & b \end{pmatrix}$ *be positive-semidefinite where we assume, without loss of generality,* $a \geq b$. *Let* $M_2^* = \begin{pmatrix} 1 & m^* \\ m^* & 1 \end{pmatrix}$ *be an optimal mask for* $z_{linx}(C_2, 1; M_2)$. *We have,*

$$\begin{aligned} &(i) \quad if\ c = 0,\ then\ m^*\ is\ any\ value\ in\ [-1, 1]; \\ &(ii) \quad if\ \tfrac{ab-1}{c^2} \geq 1,\ then\ m^* = \pm 1; \\ &(iii) \quad if\ \tfrac{ab-1}{c^2} \leq 0,\ then\ m^* = 0; \\ &(iv) \quad if\ 0 < \tfrac{ab-1}{c^2} < 1,\ then\ m^* = \pm\sqrt{\tfrac{ab-1}{c^2}}. \end{aligned}$$

Proof. Let $M_2 := \begin{pmatrix} 1 & m \\ m & 1 \end{pmatrix} \in \mathcal{M}_2$. Let $m^* = \arg\min_{-1\leq m\leq 1}\left\{(c^2m^2 + 1 - ab)^2\right\}$. Considering that $x_1 + x_2 = 1$, we obtain

$$\begin{aligned} &\text{ldet}((C_2 \circ M_2)\text{Diag}(x)(C_2 \circ M_2) + I_2 - \text{Diag}(x)) \\ =& \log((c^2m^2 + 1 - ab)^2 x_1x_2 + (ax_1 + bx_2)^2) \\ \geq& \log((c^2(m^*)^2 + 1 - ab)^2 x_1x_2 + (ax_1 + bx_2)^2) \\ =& \text{ldet}((C_2 \circ M_2^*)\text{Diag}(x)(C_2 \circ M_2^*) + I_2 - \text{Diag}(x)), \end{aligned}$$

which implies that M_2^* is an optimal mask.

The values of m^* in cases $(i - iv)$ can be easily obtained from $m^* = \arg\min_{-1\leq m\leq 1}\left\{(c^2m^2 + 1 - ab)^2\right\}$.

□

For simplicity of the following discussions, we introduce the next lemma.

Lemma 5.10. *With the same hypotheses and notations as Theorem 5.9, define* $g(a, b, c) := \exp(2z_{linx}(C_2, 1; M_2^*))$ *and*

$$\Delta z_{linx}(C, s; M) := z_{linx}(C, s) - z_{linx}(C, s; M).$$

Then

$$\Delta z_{linx}(C_2, 1; M_2^*) \geq \tfrac{1}{2}\log\tfrac{(c^2+1-ab)^2+(a+b)^2}{4g(a,b,c)}.$$

Proof. Let $\lambda_1 \geq \lambda_2 \geq 0$ be the two eigenvalues of C_2. Considering that $\lambda_1 + \lambda_2 = a + b$ and $\lambda_1\lambda_2 = ab - c^2$, the result follows directly from Lemma 5.8. □

Note that for cases $(i - ii)$ in Theorem 5.9, there is no mask better than J_2. So we focus on cases $(iii - iv)$. For case (iv), we can calculate from the proof of Theorem 5.9 that $z_{\text{linx}}(C_2, 1; M_2^*) = \frac{1}{2}\log(a^2)$. By Lemma 5.10, we have

$$\Delta z_{\text{linx}}(C_2, 1; M_2^*) \geq \tfrac{1}{2}\log\tfrac{(c^2+1-ab)^2+(a+b)^2}{4a^2}.$$

Note that $\frac{ab-1}{c^2} > 0$ and $a \geq b$ imply $a > 1$. Further, $a > 1$, $\frac{ab-1}{c^2} < 1$ and $ab \geq c^2$ imply $0 < c^2 + 1 - ab \leq 1 < a$. So,

$$\tfrac{1}{2}\log\tfrac{(c^2+1-ab)^2+(a+b)^2}{4a^2} < \tfrac{1}{2}\log\left(\tfrac{5}{4}\right).$$

Moreover, by choosing $a = b = c > 1$, we are in case (iv), and the gap becomes $\frac{1}{2}\log(1 + 1/4a^2)$, which we can make as close to $\frac{1}{2}\log(5/4)$ as we like.

For case (iii), the optimal mask is I_2, and we can find a greater gap than we could for case (iv). We prove this and our main result in the following theorem.

Theorem 5.11. *There is an infinite sequence of positive-semidefinite matrices* $\{C_n\}_{n\in 2\mathbb{Z}}$ *such that*

$$z_{linx}\left(C_n, \tfrac{n}{2}\right) - z_{linx}\left(C_n, \tfrac{n}{2}; I\right) = \tfrac{1}{4}\log\left(\tfrac{4}{3}\right)n.$$

Moreover, for $n = 2$*, this is the maximum possible lower bound on the gap that can be achieved using the lower bound from Lemma 5.8.*

Remark. *As we have indicated above in our analysis of case (iv), and proceeding similarly to how we proceed below, we can also get linear gaps with masks that are different from the identity mask, albeit with a worse constant (strictly less than* $\frac{1}{4}\log(\frac{5}{4})$*).*

Proof. (Theorem 5.11) First, consider $n = 2, s = 1$. We use the same notations as in Theorem 5.9 and consider its case (iii), where $ab \leq 1$, so that the optimal mask is I_2. In the following, we will use $\hat{x}$ to denote the optimal solution of (masked scaled linx) for $C := C_2$, $s = 1$, $M := I_2$, and $\gamma = 1$; so $z_{\text{linx}}(C_2, 1; I_2) = z_{\text{linx}}(C_2, 1; I_2, 1) = f_{\text{linx}}(C_2, 1; I_2, 1; \hat{x})$.

We have two sub-cases to analyze:

(i) $a \geq 1 \geq b$: by Theorem 5.7, $\hat{x} = (1, 0)^{\mathsf{T}}$ is an optimal solution and $z_{\text{linx}}(C_2, 1; I_2) = \frac{1}{2}\log(a^2)$. By Lemma 5.10, we have

$$\Delta z_{\text{linx}}(C_2, 1; I_2) \geq \tfrac{1}{2}\log\tfrac{(c^2+1-ab)^2+(a+b)^2}{4a^2}. \tag{5.8}$$

Note that $(c^2 + 1 - ab)^2 + (a + b)^2 \leq 5a^2$, so $\frac{(c^2+1-ab)^2+(a+b)^2}{4a^2} \leq \frac{5}{4}$. The equality can be obtained when $a = b = c = 1$.

(ii) $1 > a \geq b$: there are still two sub-cases:

$$\tfrac{1}{b^2-1} - \tfrac{1}{a^2-1} \geq 1 \text{ and } \tfrac{1}{b^2-1} - \tfrac{1}{a^2-1} < 1.$$

- If $\frac{1}{b^2-1} - \frac{1}{a^2-1} \geq 1$, then by Theorem 5.7, we also have $\hat{x} = (1, 0)^{\mathsf{T}}$ and $z_{\text{linx}}(C_2, 1; I_2) = \frac{1}{2}\log(a^2)$. Thus, (5.8) also holds. From $\frac{1}{b^2-1} - \frac{1}{a^2-1} \geq 1$ and $b^2 \geq 0$, we can see that $\frac{1}{2} \leq a^2 < 1$ and $b \leq \sqrt{2 - \frac{1}{a^2}}$. Together with $c^2 \leq ab < 1$, letting $t := \frac{1}{a^2} \in (1, 2]$, we get

$$\begin{aligned}
&\max \tfrac{(c^2+1-ab)^2+(a+b)^2}{4a^2} = \max \tfrac{1+\left(a+\sqrt{2-\frac{1}{a^2}}\right)^2}{4a^2} \\
&= \max \tfrac{1}{4} + \tfrac{3}{4}t - \tfrac{1}{4}t^2 + \tfrac{1}{2}\sqrt{2t - t^2} \\
&\leq \max \tfrac{1}{4} + \tfrac{3}{4}t - \tfrac{1}{4}t^2 + \max \tfrac{1}{2}\sqrt{2t - t^2} \\
&= \tfrac{13}{16} + \tfrac{1}{2} < \tfrac{4}{3}.
\end{aligned}$$

In the next sub-case, we will build a $\frac{1}{2}\log\left(\frac{4}{3}\right)$ gap, so the gap in the present sub-case is sub-optimal.

- If $\frac{1}{b^2-1} - \frac{1}{a^2-1} < 1$, then by Theorem 5.7, $\hat{x} = \frac{1}{2}\left(1 + \frac{1}{b^2-1} - \frac{1}{a^2-1}, 1 - \frac{1}{b^2-1} + \frac{1}{a^2-1}\right)^{\mathsf{T}}$

is an optimal solution, and we have

$$z_{\text{linx}}(C_2, 1; I_2) = \frac{1}{2}\log\left(\frac{1}{4}\left(a^2 + \frac{a^2-1}{b^2-1}\right)\left(b^2 + \frac{b^2-1}{a^2-1}\right)\right).$$

By Lemma 5.10, we have

$$\Delta z_{\text{linx}}(C_2, 1; I_2) \geq \frac{1}{2}\log\frac{(c^2+1-ab)^2+(a+b)^2}{\left(a^2+\frac{a^2-1}{b^2-1}\right)\left(b^2+\frac{b^2-1}{a^2-1}\right)}.$$

We claim that

$$\frac{(c^2+1-ab)^2+(a+b)^2}{\left(a^2+\frac{a^2-1}{b^2-1}\right)\left(b^2+\frac{b^2-1}{a^2-1}\right)} \leq \frac{1+(a+b)^2}{\left(a^2+\frac{a^2-1}{b^2-1}\right)\left(b^2+\frac{b^2-1}{a^2-1}\right)} \leq \frac{4}{3}.$$

The first inequality holds because $c^2 \leq ab < 1$ and the second holds for being equivalent to

$$(1-2ab)^2 + (a-b)^2 + 4(a^2-b^2)\left(\frac{1}{b^2-1} - \frac{1}{a^2-1}\right) \geq 0,$$

We get equality in both with $a = b = c = \frac{\sqrt{2}}{2}$.

In the analysis above, we see that we can create the largest gap in the last case. Therefore, we define

$$C_2 := \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}.$$

Then $\lambda_1 = \sqrt{2}, \lambda_2 = 0$, the optimal solution for $C_2 \circ I_2$ is $\left(\frac{1}{2}, \frac{1}{2}\right)^{\mathsf{T}}$, and

$$\begin{aligned}
&z_{\text{linx}}(C_2, 1) - z_{\text{linx}}(C_2, 1; I_2) \\
&= \frac{1}{2}\log\left(\left(\frac{1}{2}\cdot(\sqrt{2})^2 + 1 - \frac{1}{2}\right)\left(\frac{1}{2}\cdot 0^2 + 1 - \frac{1}{2}\right)\right) \\
&- \frac{1}{2}\log\left(\left(\frac{1}{2}\cdot\left(\frac{\sqrt{2}}{2}\right)^2 + 1 - \frac{1}{2}\right)\left(\frac{1}{2}\cdot\left(\frac{\sqrt{2}}{2}\right)^2 + 1 - \frac{1}{2}\right)\right) \\
&= \frac{1}{2}\log\left(\frac{4}{3}\right).
\end{aligned}$$

For $n = 2k$, we construct a block-diagonal matrix C_n with $k = \frac{n}{2}$ blocks, and each block is such a C_2 matrix. Then we take $s = \frac{n}{2}$. In this way, C_n has k eigenvalues of $\sqrt{2}$ and k eigenvalues of 0. Also, all diagonal elements of C_n are $\frac{\sqrt{2}}{2}$. By (5.7), we have

$$\begin{aligned}
&z_{\text{linx}}\left(C_n, \tfrac{n}{2}\right) - z_{\text{linx}}\left(C_n, \tfrac{n}{2}; I\right) \\
&= \frac{n}{4}\log\left(\left(\frac{1}{2}(\sqrt{2})^2 + 1 - \frac{1}{2}\right)\left(\frac{1}{2}(0)^2 + 1 - \frac{1}{2}\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&- \tfrac{n}{4}\log\left(\left(\tfrac{1}{2}\left(\tfrac{\sqrt{2}}{2}\right)^2 + 1 - \tfrac{1}{2}\right)\left(\tfrac{1}{2}\left(\tfrac{\sqrt{2}}{2}\right)^2 + 1 - \tfrac{1}{2}\right)\right) \\
&= \tfrac{1}{4}\log\left(\tfrac{4}{3}\right) n.
\end{aligned}
$$

□

Remark. *It is easy to check that with respect to the sequence of C_n in the proof of Theorem 5.11, we have $z_{linx}\left(C_n, \frac{n}{2}\right) = \frac{n}{2}\log\left(\frac{\sqrt{2}}{2}\right)$. From this and the other calculations in the proof, we can further calculate*

$$
\frac{z_{linx}\left(C_n, \frac{n}{2}; I\right) - z_{linx}\left(C_n, \frac{n}{2}\right)}{z_{linx}\left(C_n, \frac{n}{2}\right) - z_{linx}\left(C_n, \frac{n}{2}\right)} = \frac{\log\left(\frac{9}{8}\right)}{\log\left(\frac{3}{2}\right)} \approx 0.29
$$

So we can get a 71% reduction in the integrality gap, for all of the instances in this sequence, by masking.

5.3 Optimal scaling parameter: some special cases and general behavior

In this section, we first show how an appropriate scaling parameter γ can help improve the linx bound by forcing one optimal solution of scaled linx to lie in $\{0,1\}^n$ when C is diagonal or C is non-singular of order 2. Next, we show the following results: (i) if $s < \text{rank}(C)$, then an optimal scaling parameter γ for scaled linx can always be obtained, (ii) if $s = \text{rank}(C)$ and $\hat{\gamma}$ is an optimal scaling parameter for linx, then so is any $\gamma \geq \hat{\gamma}$, (iii) if $s > \text{rank}(C)$, there is no optimal γ. In fact, in this case we show that the linx-bound has the nice property of recognizing the behavior of MESP; it tends to minus infinity as γ tends to infinity.

Proposition 5.12. *For diagonal positive-definite matrix $C := \text{Diag}\{d_1, ..., d_n\}$, where $d_1 \geq ... \geq d_n > 0$ and $0 < s < n$, the scaling parameter $\hat{\gamma} = \frac{1}{d_s^2}$ forces an optimal solution of scaled linx to lie in $\{0,1\}^n$. Therefore $\hat{\gamma}$ is an optimal scaling parameter.*

Proof. Note that

$$
f_{\text{linx}}(C, s; \gamma; x) = \tfrac{1}{2}\log\prod_{i=1}^{n}(\gamma d_i^2 x_i + 1 - x_i) - \tfrac{1}{2}s\log\gamma.
$$

Partition N as $N = L' \cup E' \cup G'$, where $\gamma d_i^2 < 1, i \in L'$; $\gamma d_i^2 = 1, i \in E'$; $\gamma d_i^2 > 1, i \in G'$. As we have seen in Lemma 5.3, $f_{\text{linx}}(C, s; \gamma; x)$ strictly decreases with $x_i, i \in L'$, does not change with $x_i, i \in E'$ and strictly increases with $x_i, i \in G'$. So, if there is a $\gamma > 0$ such that $|G'| \leq s$

while $|E' \cup G'| \geq s$. By Theorem 5.7, $(e_s^\mathsf{T}, 0)^\mathsf{T}$ is an optimal solution for scaled linx which lies in $\{0,1\}^n$. In fact, $\hat{\gamma} := \frac{1}{d_s^2}$ is such a scaling parameter. Therefore, by Proposition 5.1, $\hat{\gamma}$ is optimal. □

Proposition 5.13. *Let* $C_2 := \begin{pmatrix} a & c \\ c & b \end{pmatrix}$ *be positive-definite where we assume, without loss of generality,* $a \geq b$. *Let* $s = 1$. *Then the scaling parameter* $\hat{\gamma} = \frac{a^2-c^2}{(ab-c^2)^2}$ *forces an optimal solution of scaled linx to lie in* $\{0,1\}^2$. *Therefore* $\hat{\gamma}$ *is an optimal scaling parameter.*

Proof. We have

$$\begin{aligned} & f_{\text{linx}}(C_2, 1; x) \\ &= \tfrac{1}{2}\,\text{ldet}(C_2\,\text{Diag}(x)C_2 + I_2 - \text{Diag}(x)) \\ &= \tfrac{1}{2}\log((c^2+1-ab)^2 x_1 x_2 + (ax_1 + bx_2)^2). \end{aligned}$$

Because $f_{\text{linx}}(C_2, 1; x)$ is concave, and the null space of $\mathbf{e}_2$ is $\{(t,-t)^\mathsf{T} \ : \ t \in \mathbb{R}\}$, to prove that one optimal solution lies in $\{0,1\}^2$ (in particular, we assume this optimal solution is $\hat{x} = (1,0)^\mathsf{T}$), we only need to prove

$$\left.\frac{f_{\text{linx}}(C_2, 1; \hat{x} - t(e_1 - e_2))}{\partial t}\right|_{t=0} \leq 0$$

which is equivalent to

$$\frac{\partial f_{\text{linx}}(C_2, 1; \hat{x})}{\partial x_1} - \frac{\partial f_{\text{linx}}(C_2, 1; \hat{x})}{\partial x_2} \geq 0, \tag{5.9}$$

and finally $2(a^2 - c^2) - (c^2 - ab)^2 - 1 \geq 0$.

Because $f_{\text{linx}}(C_2, s; \gamma; x) = f_{\text{linx}}(\sqrt{\gamma}C_2, s; x) + \frac{1}{2}\log\gamma$, we have that

$$\frac{\partial f_{\text{linx}}(C_2, 1; \gamma; \hat{x})}{\partial x_1} - \frac{\partial f_{\text{linx}}(C_2, 1; \gamma; \hat{x})}{\partial x_2} \geq 0,$$

is equivalent to

$$\frac{\partial f_{\text{linx}}(\sqrt{\gamma}C_2, 1; \hat{x})}{\partial x_1} - \frac{\partial f_{\text{linx}}(\sqrt{\gamma}C_2, 1; \hat{x})}{\partial x_2} \geq 0,$$

and finally

$$2(a^2 - c^2)\gamma - (c^2 - ab)^2\gamma^2 - 1 \geq 0. \tag{5.10}$$

The left-hand side of (5.10) is maximized by $\hat{\gamma} = \frac{a^2-c^2}{(ab-c^2)^2}$ and the corresponding value is $\frac{(a^2-c^2)^2}{(ab-c^2)^2} - 1$ which is nonnegative because $a^2 \geq ab > c^2$. Note that if $a > b$ then $\frac{(a^2-c^2)^2}{(ab-c^2)^2} - 1 > 0$, which means there is an $\epsilon > 0$ such that for any $\gamma \in [\hat{\gamma} - \epsilon, \hat{\gamma} + \epsilon]$, $\frac{(a^2-c^2)^2}{(ab-c^2)^2} - 1 \geq 0$ and $\hat{x}$ is an

optimal solution, i.e., the optimal scaling parameter is not unique. Finally, by Proposition 5.1, $\hat{\gamma}$ is optimal. □

Not unexpectedly, there also exists a large and simple class of C where no optimal solution of scaled linx lies in $\{0,1\}^n$ for any $0 < s < n$ and any scaling parameter γ, as we will see in Theorem 5.15.

First, note that when $C = \tau_1 I$, for any $\tau_1 > 0$, in all cases of Theorem 5.7, $\hat{x} := \frac{s}{n}\mathbf{e}$ is an optimal solution of linx. The same observation can be extracted from Lemma 5.2. In fact, this observation is also a special case of the following result, which follows immediately from the concavity of $f_{\text{linx}}(C, s; x)$ and its symmetry in this case.

Proposition 5.14. *Suppose that $\tau_1 > 0$, $\tau_2 \geq 0$, and $0 < s < n$ integer. Let $C = \tau_1 I + \tau_2 J$, then $\hat{x} = \frac{s}{n}\mathbf{e}$ is an optimal solution for linx.*

Theorem 5.15. *For any order $n \geq 3$, any $0 < s < n$ and $C = \tau_1 I + \tau_2 J$, $\tau_1 > 0$, $\tau_2 > 0$, and for any scaling parameter $\gamma > 0$, the optimal solution of scaled linx cannot lie in $\{0,1\}^n$.*

Proof. By Proposition 5.14, one optimal solution for scaled linx is $\frac{s}{n}\mathbf{e}$ under the setting of this theorem. From the proof of [Theorem 21, (Chen, Fampa, Lambert, and Lee, 2021)], we have that if

$$\begin{aligned} &\gamma C \operatorname{Diag}(y) C - \operatorname{Diag}(y) \neq 0, \\ &\text{when } \mathbf{e}^{\mathsf{T}} y = 0,\ -\mathbf{e} \leq y \leq \mathbf{e},\ y \neq 0, \end{aligned} \tag{5.11}$$

then $f_{\text{linx}}(C, s; \gamma; x)$ is strictly concave with a unique optimal solution on the feasible region of scaled linx. Because $\frac{s}{n}\mathbf{e}$ is already optimal in this case, we see that the optimal solution cannot lie in $\{0,1\}^n$. Now we prove that (5.11) holds.

Substituting $\tau_1 I + \tau_2 J$ for C in (5.11) and dividing by γ, we get

$$\left(\tau_1^2 - \tfrac{1}{\gamma}\right) \operatorname{Diag}(y) + \tau_1 \tau_2 (\mathbf{e} y^{\mathsf{T}} + y \mathbf{e}^{\mathsf{T}}) \neq 0 \tag{5.12}$$

It is easy to see that if (5.12) is *not* satisfied, then $y_i + y_j = 0$ for all $i \neq j$. But this cannot be true when $n \geq 3$ for $y \neq 0$. □

Interestingly, there is also a very simple example for which no $\gamma > 0$ can be an optimal scale factor:

Proposition 5.16. *For $C := J_2$, $s := 1$, there is no optimal scaling factor γ for scaled linx. In fact, for all $\gamma > 0$,*

$$z_{linx}(J_2, 1; \gamma) = \tfrac{1}{2} \log\left(1 + \tfrac{1}{4\gamma}\right).$$

which monotonically decreases as γ increases.

Proof.

$$\begin{aligned}
&z_{\text{linx}}(J_2, 1; \gamma) \\
&= \tfrac{1}{2} \max_{\substack{x_1+x_2=1 \\ 0\le x_1,x_2\le 1}} \{\log\left(\gamma(x_1+x_2)(2-x_1-x_2)+(1-x_1)(1-x_2)\right) - \log\gamma\} \\
&\le \tfrac{1}{2} \max_{\substack{x_1+x_2=1 \\ 0\le x_1,x_2\le 1}} \left\{\log\left(\gamma(x_1+x_2)(2-x_1-x_2)+\left(\tfrac{2-x_1-x_2}{2}\right)^2\right) - \log\gamma\right\} \\
&= \tfrac{1}{2}\left(\log\left(\gamma+\tfrac{1}{4}\right) - \log\gamma\right) = \tfrac{1}{2}\log\left(1+\tfrac{1}{4\gamma}\right).
\end{aligned}$$

Note that both maximums are achieved at $x_1 = x_2 = 1/2$, and so the inequality is an equation. □

Based on Proposition 5.16, our interest is in what cases, we are guaranteed to have a finite optimal scaling parameter γ. In fact, a broad sufficient condition is $s < \text{rank}(C)$ by the following theorem.

Theorem 5.17. *For all positive-semidefinite C and $0 < s < n$, we have*

$$\lim_{\gamma\to 0} z_{linx}(C, s; \gamma) = +\infty.$$

If we further assume that $s < \text{rank}(C)$, then

$$\lim_{\gamma\to +\infty} z_{linx}(C, s; \gamma) = +\infty.$$

Proof. For all $\gamma > 0$, by setting $\bar{x} := \frac{s}{n}\mathbf{e}$, we have

$$\begin{aligned}
z_{\text{linx}}(C, s; \gamma) &\ge f_{\text{linx}}(C, s; \gamma; \bar{x}) \\
&= \tfrac{1}{2}\left(\text{ldet}\left(\gamma\tfrac{s}{n}C^2 + \left(1-\tfrac{s}{n}\right)I\right) - s\log\gamma\right).
\end{aligned}$$

When $\gamma \to 0$, $\gamma\frac{s}{n}C^2 + \left(1-\frac{s}{n}\right)I \to \left(1-\frac{s}{n}\right)I$. So

$$\begin{aligned}
&\lim_{\gamma\to 0} \text{ldet}\left(\gamma\tfrac{s}{n}C^2 + \left(1-\tfrac{s}{n}\right)I\right) = n\log\left(1-\tfrac{s}{n}\right), \\
&\text{and } \lim_{\gamma\to 0} -s\log\gamma = +\infty.
\end{aligned}$$

Therefore, $\lim_{\gamma\to 0} z_{\text{linx}}(C,s;\gamma) = +\infty$.

But we can also write $f_{\text{linx}}(C,s;\gamma;\bar{x}) =$

$$\tfrac{1}{2}\left(\text{ldet}\left(\tfrac{s}{n}C^2 + \tfrac{1}{\gamma}\left(1-\tfrac{s}{n}\right)I\right) + (n-s)\log\gamma\right).$$

Note that $\lim_{\gamma\to+\infty}(n-s)\log\gamma = +\infty$. Further, if C is non-singular, then

$$\lim_{\gamma\to+\infty}\text{ldet}\left(\tfrac{s}{n}C^2 + \tfrac{1}{\gamma}\left(1-\tfrac{s}{n}\right)I\right) = \text{ldet}\left(\tfrac{s}{n}C^2\right).$$

So we can conclude that $\lim_{\gamma\to+\infty} z_{\text{linx}}(C,s;\gamma) = +\infty$, when C is nonsingular.

If C is singular, then we have

$$\lim_{\gamma\to+\infty}\text{ldet}\left(\tfrac{s}{n}C^2 + \tfrac{1}{\gamma}\left(1-\tfrac{s}{n}\right)I\right) = -\infty,$$

and we cannot immediately conclude anything useful. So we proceed differently. When $s < \text{rank}(C)$, without loss of generality, we can write $C = Q\Lambda Q^{\mathsf{T}}$, where Q is orthogonal and $\Lambda := \text{Diag}(\lambda_1, ..., \lambda_n)$ with $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$. We have $\lambda_i \neq 0$ for $i \leq \text{rank}(C)$ and $\lambda_i = 0$ for $i > \text{rank}(C)$. By L'Hôptital's rule,

$$\begin{aligned}
&\lim_{\gamma\to+\infty}\frac{\text{ldet}\left(\frac{s}{n}C^2+\frac{1}{\gamma}\left(1-\frac{s}{n}\right)I\right)}{(n-s)\log\gamma}\\
=&\lim_{\gamma\to+\infty}\frac{\partial\left(\text{ldet}\left(\frac{s}{n}C^2+\frac{1}{\gamma}\left(1-\frac{s}{n}\right)I\right)\right)/\partial\gamma}{\partial((n-s)\log\gamma)/\partial\gamma}\\
=&\lim_{\gamma\to+\infty}\frac{\text{tr}\left(\left(\frac{s}{n}C^2+\frac{1}{\gamma}\left(1-\frac{s}{n}\right)I\right)^{-1}\left(1-\frac{s}{n}\right)I\right)\frac{-1}{\gamma^2}}{(n-s)\frac{1}{\gamma}}\\
=&\lim_{\gamma\to+\infty}\frac{-1}{n\gamma}\text{tr}\left(\left(\tfrac{s}{n}C^2+\tfrac{1}{\gamma}\left(1-\tfrac{s}{n}\right)I\right)^{-1}\right)\\
=&\lim_{\gamma\to+\infty}\frac{-1}{n}\text{tr}\left(\left(\gamma\tfrac{s}{n}C^2+\left(1-\tfrac{s}{n}\right)I\right)^{-1}\right)\\
=&\lim_{\gamma\to+\infty}\frac{-1}{n}\text{tr}\left(\left(\gamma\tfrac{s}{n}Q\Lambda^2Q^{\mathsf{T}}+\left(1-\tfrac{s}{n}\right)QQ^{\mathsf{T}}\right)^{-1}\right)\\
=&\lim_{\gamma\to+\infty}\frac{-1}{n}\text{tr}\left(Q\left(\gamma\tfrac{s}{n}\Lambda^2+\left(1-\tfrac{s}{n}\right)I\right)^{-1}Q^{\mathsf{T}}\right)\\
=&\frac{-1}{n-s}\text{tr}\left(\text{Diag}\left\{\mathbf{0}_{1\times\text{rank}(C)},\mathbf{e}^{\mathsf{T}}_{n-\text{rank}(C)}\right\}\right)\\
=&-\frac{n-\text{rank}(C)}{n-s}.
\end{aligned}$$

This means for every $\epsilon > 0$, there exists $\gamma_\epsilon > 0$ such that when $\gamma > \max\{\gamma_\epsilon, 1\}$,

$$\tfrac{1}{2}\left(\text{ldet}\left(\tfrac{s}{n}C^2 + \tfrac{1}{\gamma}\left(1-\tfrac{s}{n}\right)I\right) + (n-s)\log\gamma\right)$$

$$\geq \frac{1}{2}\left(-\frac{n-\text{rank}(C)}{n-s}-\epsilon+1\right)(n-s)\log\gamma.$$

So

$$\begin{aligned}
&\lim_{\gamma\to+\infty} z_{\text{linx}}(C,s;\gamma)\\
\geq&\lim_{\epsilon\to 0}\lim_{\gamma\to+\infty}\left(-\frac{n-\text{rank}(C)}{n-s}-\epsilon+1\right)(n-s)\log\gamma\\
=&\lim_{\epsilon\to 0}\lim_{\gamma\to+\infty}\left(\frac{\text{rank}(C)-s}{n-s}-\epsilon\right)(n-s)\log\gamma\\
=&+\infty.
\end{aligned}$$

□

Corollary 5.18. *For all positive-semidefinite C and $0<s<n$ where $s<\text{rank}(C)$, we can find a finite optimal scaling parameter $\hat{\gamma}$ such that*

$$z_{linx}(C,s;\hat{\gamma})=\min_{\gamma>0} z_{linx}(C,s;\gamma).$$

Proof. By (Chen, Fampa, Lambert, and Lee, 2021), if we replace γ with e^{ψ}, then $z_{\text{linx}}(C,s;e^{\psi})$ is convex and continuous in ψ and by Theorem 5.17,

$$\begin{aligned}
\lim_{\psi\to-\infty} z_{\text{linx}}(C,s;e^{\psi})=\lim_{\gamma\to 0} z_{\text{linx}}(C,s;\gamma)=+\infty\\
\lim_{\psi\to+\infty} z_{\text{linx}}(C,s;e^{\psi})=\lim_{\gamma\to+\infty} z_{\text{linx}}(C,s;\gamma)=+\infty.
\end{aligned}$$

We can conclude that a minimizing $\hat{\psi}$ exists, and then we have the minimizer $\hat{\gamma}:=e^{\hat{\psi}}$. □

When $s=\text{rank}(C)$, the following result establishes that $\lim_{\gamma\to\infty} z_{\text{linx}}(C,s;\gamma)$ exists and is finite, and $z_{\text{linx}}(C,s;\gamma)$ is monotonically non-increasing in γ. This implies that if $\hat{\gamma}>0$ is optimal, then all $\gamma\geq\hat{\gamma}$ are optimal.

Theorem 5.19. *When $s=\text{rank}(C)$, without loss of generality, we can write C as $C=Q\Lambda Q^{\mathsf{T}}$, where Q is orthogonal and $\Lambda:=\text{Diag}(\lambda_1,...,\lambda_n)$ with $\lambda_1\geq...\geq\lambda_s>\lambda_{s+1}=...=\lambda_n=0$. Denote $\Lambda_s:=\text{Diag}(\lambda_1,...,\lambda_s)$. Denote $P=Q^{\mathsf{T}}\,\text{Diag}(x)Q$, P_s as the principal sub-matrix of P indexed by $(1,...,s)$, P_{n-s} as the principal sub-matrix of P indexed by $(s+1,...,n)$ and $P_{s,n-s}$ as the sub-matrix of P with rows indexed by $(1,...,s)$ and columns indexed by $(s+1,...,n)$ ($P_{n-s,s}$ similarly).*

Then the value $\lim_{\gamma\to+\infty} z_{linx}(C,s;\gamma)$ exists and is the optimal value of the following convex

program:

$$\begin{array}{ll} \max & \frac{1}{2}(\text{ldet}\,(\Lambda_s P_s \Lambda_s) + \text{ldet}\,(I_{n-s} - P_{n-s})) \\ s.t. & \mathbf{e}^\mathsf{T} x = s \\ & 0 \le x \le 1. \end{array} \tag{5.13}$$

Furthermore, $z_{linx}(C, s; \gamma)$ is monotonically non-increasing in γ.

Proof. By the conditions,

$$\begin{aligned} &\text{ldet}(\gamma C \,\text{Diag}(x) C + I - \text{Diag}(x)) - s \log \gamma \\ &= \text{ldet}(\gamma \Lambda P \Lambda + I - P) - s \log \gamma. \end{aligned}$$

Because C, s are fixed, let $F_s(\gamma; x) := \gamma \Lambda_s P_s \Lambda_s + I_s - P_s$ be the principal sub-matrix of $\gamma \Lambda P \Lambda + I - P$ indexed by $(1, ..., s)$. We first prove that for any $\gamma > 0$ and any x feasible, $F_s(\gamma; x)$ is positive-definite so that we can use Schur complement formula to represent the determinant of $\gamma \Lambda P \Lambda + I - P$.

The construction of P implies its eigenvalues are $\{x_1, x_2, ..., x_n\}$ so all eigenvalues of P lie in $[0, 1]$. Because P_s is a principal sub-matrix of P, by [Theorem 4.3.17, (Horn and Johnson, 1985)], all eigenvalues of P_s lie in $[0, 1]$. Decompose P_s as $P_s = \hat{Q} \hat{\Lambda} \hat{Q}^\mathsf{T}$ where $\hat{Q}$ is orthogonal and $\hat{\Lambda}$ is the diagonal matrix of eigenvalues of P_s. In particular, all elements of $\text{diag}(\hat{\Lambda})$ are in $[0, 1]$. Let $\hat{C} = \hat{Q}^\mathsf{T} \Lambda_s \hat{Q}$, then

$$F_s(\gamma; x) = \hat{Q} \left(\gamma \hat{C} \hat{\Lambda} \hat{C} + I_s - \hat{\Lambda} \right) \hat{Q}^\mathsf{T}.$$

Because Λ_s is positive-definite, so is $\hat{C}$. By [Lemma 20, (Chen, Fampa, Lambert, and Lee, 2021)], $F_s(\gamma; x)$ is positive-definite for any $\hat{\Lambda}$ where $0 \le \text{diag}(\hat{\Lambda}) \le \mathbf{e}$.

We only need to consider x in the feasible region such that $\gamma \Lambda P \Lambda + I - P$, (equivalently, $\gamma C \,\text{Diag}(x) C + I - \text{Diag}(x)$) is positive-definite. So we assume that $\gamma \Lambda P \Lambda + I - P$ is positive-definite in the following. Then the Schur complement of $\gamma \Lambda P \Lambda + I - P$ in $F_s(\gamma; x)$, which is $I_{n-s} - P_{n-s} - P_{n-s,s} F_s(\gamma; x)^{-1} P_{s,n-s}$, is also positive-definite. Furthermore, $P_{n-s,s} F_s(\gamma; x)^{-1} P_{s,n-s}$ is positive-semidefinite by the positive-definiteness of $F_s(\gamma; x)$ and we get that $I_{n-s} - P_{n-s}$ is positive-definite. On the other hand, because the feasible region of x is compact, and the objective value of (5.13) is upper bounded, the optimal value of (5.13) is attainable by some x such that the corresponding $\Lambda_s P_s \Lambda_s$ and $I_{n-s} - P_{n-s}$ are positive-definite. So we justify the definition of (5.13), and

$$\begin{aligned} &\text{ldet}(\gamma \Lambda P \Lambda + I - P) - s \log \gamma \\ &= \text{ldet}(F_s(\gamma; x)) - s \log \gamma \end{aligned}$$

$$
\begin{aligned}
&+\mathrm{ldet}(I_{n-s}-P_{n-s}-P_{n-s,s}F_s(\gamma;x)^{-1}P_{s,n-s})\\
&=\mathrm{ldet}\left(\Lambda_s P_s\Lambda_s+\tfrac{1}{\gamma}(I_s-P_s)\right)\\
&+\mathrm{ldet}(I_{n-s}-P_{n-s}-P_{n-s,s}F_s(\gamma;x)^{-1}P_{s,n-s}).
\end{aligned}
$$

Denote the optimal solution of (5.13) as x^* and $P^* = Q^{\mathsf{T}}\,\mathrm{Diag}(x^*)Q$. We claim that $\lim_{\gamma\to+\infty} z_{\mathrm{linx}}(C,s;\gamma) =$

$$
\tfrac{1}{2}\left(\mathrm{ldet}(\Lambda_s P_s^*\Lambda_s)+\mathrm{ldet}(I_{n-s}-P_{n-s}^*)\right).
$$

We now prove this claim. In fact, for any x feasible to scaled linx such that $\gamma\Lambda P\Lambda + I - P$ is positive-definite and that $F_s(\gamma;x)$ is positive-definite, we have

$$
\begin{aligned}
&\mathrm{ldet}(\Lambda_s P_s\Lambda_s+\tfrac{1}{\gamma}(I_s-P_s))\\
&+\mathrm{ldet}(I_{n-s}-P_{n-s}-P_{n-s,s}F_s(\gamma;x)^{-1}P_{s,n-s})\\
&\le \mathrm{ldet}(\Lambda_s P_s\Lambda_s+\tfrac{1}{\gamma}I_s)+\mathrm{ldet}(I_{n-s}-P_{n-s}).
\end{aligned}
\tag{5.14}
$$

We further assume that $\Lambda_s P_s\Lambda_s$ is positive-definite otherwise the right-hand-side of (5.14) goes to minus infinity as γ goes to infinity because $\mathrm{ldet}(I_{n-s}-P_{n-s})$ is clearly upper bounded by 0 for any x. Decompose $\Lambda_s P_s\Lambda_s$ as $Q'\Lambda' Q'^{\mathsf{T}}$ where Q' is orthogonal and $\Lambda' := \mathrm{Diag}(\lambda'_1,...,\lambda'_s)$ where $\lambda'_1 \ge \lambda'_2 \ge \ldots \ge \lambda'_s > 0$ is the diagonal matrix of eigenvalues of $\Lambda_s P_s\Lambda_s$. Then

$$
\mathrm{ldet}\left(\Lambda_s P_s\Lambda_s+\tfrac{1}{\gamma}I_s\right)=\log\left(\prod_{i=1}^{s}\left(\lambda'_i+\tfrac{1}{\gamma}\right)\right).
$$

Because every element of $\Lambda_s P_s\Lambda_s$ is bounded by a uniform number for any x, by Gershgorin circle theorem, $\lambda'_i, i\in\{1,...,s\}$ are bounded by a uniform number for all x. We pick a positive number $L_1 > 0$, when $\gamma \ge L_1$, there is a compact set $\mathcal{H}\subset\mathbb{R}^s$ (independent of γ) such that for all x feasible to scaled linx, $(\lambda'_1+\frac{1}{\gamma},...,\lambda'_s+\frac{1}{\gamma})^{\mathsf{T}}$ as well as $(\lambda'_1,...,\lambda'_s)^{\mathsf{T}}$ belongs to $\mathcal{H}$. Because the function $\prod_{i=1}^s y_i$ is continuous differentiable in y on $\mathbb{R}^s$, it is Lipschitz continuous on $\mathcal{H}$, then, $\exists\ L_2 > 0$ such that

$$
\left|\prod_{i=1}^{s}\left(\lambda'_i+\tfrac{1}{\gamma}\right)-\prod_{i=1}^{s}\lambda'_i\right|\le\tfrac{L_2\sqrt{s}}{\gamma}.
$$

Because $I_{n-s}-P_{n-s}$ is positive-definite and every element is bounded by a uniform number

for any x, there exists $L_3 > 0$,

$$0 < \det(I_{n-s} - P_{n-s}) \leq L_3.$$

With the above arguments, when $\gamma \geq L_1$, we have

$$\begin{aligned}
&\det\left(\Lambda_s P_s \Lambda_s + \tfrac{1}{\gamma} I_s\right) \det(I_{n-s} - P_{n-s}) \\
&= \left(\prod_{i=1}^{s} \left(\lambda'_i + \tfrac{1}{\gamma}\right)\right) \det(I_{n-s} - P_{n-s}) \\
&\leq \left(\prod_{i=1}^{s} \lambda'_i + \tfrac{L_2\sqrt{s}}{\gamma}\right) \det(I_{n-s} - P_{n-s}) \\
&= \det(\Lambda_s P_s \Lambda_s) \det(I_{n-s} - P_{n-s}) \\
&\qquad + \tfrac{L_2\sqrt{s}}{\gamma} \det(I_{n-s} - P_{n-s}) \\
&\leq \det(\Lambda_s P_s^* \Lambda_s) \det(I_{n-s} - P^*_{n-s}) \\
&\qquad + \tfrac{L_2\sqrt{s}}{\gamma} \det(I_{n-s} - P_{n-s}) \\
&\leq \det(\Lambda_s P_s^* \Lambda_s) \det(I_{n-s} - P^*_{n-s}) + \tfrac{L_2 L_3 \sqrt{s}}{\gamma}.
\end{aligned}$$

For any x such that $\Lambda_s P_s \Lambda_s$ is singular, because the eigenvalues of $\Lambda_s P_s \Lambda_s$ are upper bounded uniformly for all x feasible, clearly there is some $L_4 > 0$ such that when $\gamma \geq L_4$, any such x cannot be an optimal solution for scaled linx.

Because $\log(\cdot)$ is monotonically increasing, the above implies that when $\gamma \geq \max\{L_1, L_4\}$, we have

$$\begin{aligned}
&z_{\text{linx}}(C, s; \gamma) \\
&= \max_{\substack{\mathbf{e}^\mathsf{T} x = s, \\ 0 \leq x \leq \mathbf{e}}} \tfrac{1}{2} \Big(\operatorname{ldet}\left(\Lambda_s P_s \Lambda_s + \tfrac{1}{\gamma}(I_s - P_s)\right) \\
&\quad + \operatorname{ldet}(I_{n-s} - P_{n-s} - P_{n-s,s} F_s(\gamma; x)^{-1} P_{s,n-s}) + \tfrac{L_2 L_3 \sqrt{s}}{\gamma} \Big).
\end{aligned}$$

Taking limits on both sides, we have

$$\begin{aligned}
&\lim_{\gamma \to +\infty} z_{\text{linx}}(C, s; \gamma) \leq \\
&\lim_{\gamma \to +\infty} \tfrac{1}{2} \log\Big(\det(\Lambda_s P_s^* \Lambda_s) \det(I_{n-s} - P^*_{n-s}) + \tfrac{L_2 L_3 \sqrt{s}}{\gamma} \Big) \\
&= \tfrac{1}{2} \left(\operatorname{ldet}(\Lambda_s P_s^* \Lambda_s) + \operatorname{ldet}(I_{n-s} - P^*_{n-s}) \right).
\end{aligned}$$

On the other hand, the optimal solution x^* of (5.13) is feasible to scaled linx and we have proved before that $\Lambda_s P_s^* \Lambda_s$ and $I_{n-s} - P_{n-s}^*$ are positive-definite, we have $\lim_{\gamma\to\infty} F_s(\gamma; x^*)^{-1} = O_n$ where O_n is an all-zeros order-n matrix and $z_{\text{linx}}(C, s; \gamma) \geq f(C, s; \gamma; x^*)$. Furthermore,

$$\begin{aligned}
&\lim_{\gamma\to+\infty} z_{\text{linx}}(C, s; \gamma) \geq \lim_{\gamma\to+\infty} f(C, s; \gamma; x^*) \\
&= \lim_{\gamma\to+\infty} \tfrac{1}{2} \Big(\operatorname{ldet}\Big(\Lambda_s P_s^* \Lambda_s + \tfrac{1}{\gamma}(I_s - P_s^*) \Big) \\
&+ \operatorname{ldet}(I_{n-s} - P_{n-s}^* - P_{n-s,s}^* F_s(\gamma; x^*)^{-1} P_{s,n-s}^*) \Big) \\
&= \tfrac{1}{2} \left(\operatorname{ldet}(\Lambda_s P_s^* \Lambda_s) + \operatorname{ldet}(I_{n-s} - P_{n-s}^*) \right).
\end{aligned}$$

In all, we have $\lim_{\gamma\to+\infty} z_{\text{linx}}(C, s; \gamma)$

$$= \tfrac{1}{2}\big(\operatorname{ldet}(\Lambda_s P_s^* \Lambda_s) + \operatorname{ldet}(I_{n-s} - P_{n-s}^*) \big).$$

Finally, because $z_{\text{linx}}(C, s; e^\psi)$ is convex in ψ (see [Theorem 18, (Chen, Fampa, Lambert, and Lee, 2021)]) and has a finite limit as $\psi \to +\infty$, we can conclude that $z_{\text{linx}}(C, s; e^\psi)$ is non-increasing in ψ, and hence $z_{\text{linx}}(C, s; \gamma)$ is non-increasing in γ. □

At the outset, we assumed $s \leq \operatorname{rank}(C)$. Of course, the case where $s > \operatorname{rank}(C)$ is a bit strange because the optimal value of MESP is always $-\infty$. But by the following theorem, the linx-bound problem can recognize these cases.

Theorem 5.20. *If $s > \operatorname{rank}(C)$, then $\lim_{\gamma\to+\infty} z_{linx}(C, s; \gamma) = -\infty$, and there is no optimal γ.*

Proof. Let $r = \operatorname{rank}(C)$. We use similar notations as in Theorem 5.19, but with the a little difference. Here we have $\Lambda_r := \operatorname{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_r)$ and P_r, P_{n-r}, $P_{n-r,r}$, $P_{r,n-r}$ similarly because $r < s$. Then

$$\begin{aligned}
&\operatorname{ldet}(\gamma C \operatorname{Diag}(x) C + I - \operatorname{Diag}(x)) - s \log \gamma \\
&= \operatorname{ldet}\Big(\Lambda_r P_r \Lambda_r + \tfrac{1}{\gamma}(I_r - P_r) \Big) - (s - r) \log \gamma \\
&\quad + \operatorname{ldet}(I_{n-r} - P_{n-r} - P_{n-r,r} F_r(\gamma; x)^{-1} P_{r,n-r}).
\end{aligned}$$

We consider the convex program

$$\begin{aligned}
&\max\ \tfrac{1}{2} \left(\operatorname{ldet}\left(\Lambda_r P_r \Lambda_r \right) + \operatorname{ldet}(I_{n-r} - P_{n-r}) \right) \\
&\text{s.t. } \mathbf{e}^\mathsf{T} x = s,\ 0 \leq x \leq 1.
\end{aligned} \tag{5.15}$$

Similar to that in Theorem 5.19, (5.15) is well-defined. Denote the optimal solution of (5.15) as x^* and we have corresponding P_r^* and P_{n-r}^*, by similar arguments as in Theorem 5.19, there exists $L_1, L_2, L_3, L_4 > 0$ such that when $\gamma \geq \max\{L_1, L_4\}$,

$$\begin{aligned}
&\lim_{\gamma\to+\infty} z_{\text{linx}}(C, s; \gamma) \\
&= \lim_{\gamma\to+\infty} \max_{\substack{\mathbf{e}^\mathsf{T}x=s,\\ 0\leq x\leq \mathbf{e}}} \tfrac{1}{2}\Big(\operatorname{ldet}\Big(\Lambda_r P_r \Lambda_r + \tfrac{1}{\gamma}(I_r - P_r)\Big) \\
&\quad + \operatorname{ldet}(I_{n-r} - P_{n-r} - P_{n-r,r} F_r(\gamma; x)^{-1} P_{r,n-r}) \\
&\quad - (s-r)\log\gamma\Big) \\
&\leq \lim_{\gamma\to+\infty} \tfrac{1}{2}\Big(\operatorname{ldet}(\Lambda_r P_r^* \Lambda_r) + \operatorname{ldet}(I_{n-r} - P_{n-r}^*) \\
&\quad + \tfrac{L_2 L_3 \sqrt{r}}{\gamma} - (s-r)\log\gamma\Big) = -\infty.
\end{aligned}$$

□

5.4 Linear gap under optimal scaling

In Theorem 5.11, we constructed an infinite sequence $\{C_n\}_{n\in 2\mathbb{Z}}$ where by choosing mask I, we decreased the linx bound by an amount that is at least linear in n (specifically, $\approx .0312n$). This is even the case when we choose optimal scaling parameters γ (separately), with some sacrifice in the constant.

Theorem 5.21. *There is an infinite sequence of positive-semidefinite matrices $\{C_n\}_{n\in 4\mathbb{Z}}$, such that*

$$\min_{\gamma>0} z_{linx}\left(C_n, \tfrac{n}{2}; \gamma\right) - \min_{\bar{\gamma}>0} z_{linx}\left(C_n, \tfrac{n}{2}; I, \bar{\gamma}\right) \geq bn$$

for some positive scalar $b \geq 0.024036$.

Proof. We consider a crafted sequence of C_n. Assuming $n = 4k$, and C_n is block diagonal with k blocks as $\begin{pmatrix} 1 & c_1 \\ c_1 & 1 \end{pmatrix}$ and k blocks as $\begin{pmatrix} 1 & c_2 \\ c_2 & 1 \end{pmatrix}$ where $c_1 \neq c_2$, $c_1^2 \leq 1$, $c_2^2 \leq 1$.

By Lemma 5.8,

$$\begin{aligned}
&z_{\text{linx}}\left(C_n, \tfrac{n}{2}; \gamma\right) \\
&\geq \sum_{i=1}^{k} \Big(\tfrac{1}{2}\log\Big(\tfrac{(1-c_1^2)^2}{4}\gamma + \tfrac{1+c_1^2}{2} + \tfrac{1}{4\gamma}\Big)
\end{aligned}$$

$$
\begin{aligned}
&\quad + \tfrac{1}{2}\log\left(\tfrac{(1-c_2^2)^2}{4}\gamma + \tfrac{1+c_2^2}{2} + \tfrac{1}{4\gamma}\right)\Big) \\
&= k\left(\tfrac{1}{2}\log\left(\tfrac{(1-c_1^2)^2}{4}\gamma + \tfrac{1+c_1^2}{2} + \tfrac{1}{4\gamma}\right)\right. \\
&\quad \left. + \tfrac{1}{2}\log\left(\tfrac{(1-c_2^2)^2}{4}\gamma + \tfrac{1+c_2^2}{2} + \tfrac{1}{4\gamma}\right)\right).
\end{aligned}
$$

If $c_1^2, c_2^2 < 1$, the minimum of $\frac{(1-c_1^2)^2}{4}\gamma + \frac{1+c_1^2}{2} + \frac{1}{4\gamma}$ is 1, achieved by the unique minimizer $\hat{\gamma}_1 = 1 - c_1^2$, and the minimum of $\frac{(1-c_2^2)^2}{4}\gamma + \frac{1+c_2^2}{2} + \frac{1}{4\gamma}$ is 1, achieved by the unique minimizer $\hat{\gamma}_2 = 1 - c_2^2$. If $c_1^2 = 1$, then no matter what value γ is, $\frac{(1-c_2^2)^2}{4}\gamma + \frac{1+c_2^2}{2} + \frac{1}{4\gamma}$ is always greater than 1. The case for $c_2^2 = 1$ is similar.

Thus we can choose $c_1^2 \neq c_2^2$, then for all possible values of c_1, c_2,

$$
b_{c_1,c_2} := \min_{\gamma>0} \sum_{i=1}^{2} \log\Big(\tfrac{(1-c_i^2)^2}{4}\gamma + \tfrac{1+c_i^2}{2} + \tfrac{1}{4\gamma}\Big) > 0. \tag{5.16}
$$

Then we have

$$
\min_{\gamma>0} z_{\text{linx}}\left(C_n, \tfrac{n}{2}; \gamma\right) \geq \tfrac{1}{2}k b_{c_1,c_2} = \tfrac{b_{c_1,c_2}}{8} n.
$$

On the other hand, by Proposition 5.12, we have $\min_{\bar{\gamma}} z_{\text{linx}}\left(C_n, \frac{n}{2}; I, \bar{\gamma}\right) = z_{\text{linx}}\left(C_n, \frac{n}{2}; I, 1\right) = 0$. So,

$$
\begin{aligned}
&\min_{\gamma} z_{\text{linx}}\left(C_n, \tfrac{n}{2}; \gamma\right) - \min_{\bar{\gamma}} z_{\text{linx}}\left(C_n, \tfrac{n}{2}; I, \bar{\gamma}\right) \\
&\quad \geq \tfrac{b_{c_1,c_2}}{8} n.
\end{aligned}
$$

Letting $b := \frac{b_{c_1,c_2}}{8}$, we get what we want. In particular, if we set $c_1 := 0, c_2 := 1$, the optimal γ for (5.16) is $\hat{\gamma} = \frac{1+\sqrt{3}}{2}$ and $\frac{b_{0,1}}{8} =$

$$
\begin{aligned}
&\tfrac{1}{8}\Big(\log\left(1 + \tfrac{1}{2(1+\sqrt{3})}\right) + \log\left(\tfrac{1}{2} + \tfrac{1}{2(1+\sqrt{3})} + \tfrac{1+\sqrt{3}}{8}\right)\Big) \\
&\quad \geq 0.024036.
\end{aligned}
$$

□

Remark. *It is easy to check that with respect to the sequence of C_n in the proof of Theorem 5.21, we have $z(C_n, \frac{n}{2}) = 0$. From this and other calculations in the proof, we can see that with masking, we fully close the integrality gap, for all of instances in this sequence.*

5.5 Concluding remarks

For a positive integer s, let $n := 2s$. Now, for all $n' \geq n$, consider an $n' \times n'$ block-diagonal matrix, with one diagonal block being C_n (from Theorem 5.21 or 5.11), and another diagonal block being $\epsilon I_{n'-n}$, for small $\epsilon > 0$. Then, using Lemma 5.6, it can be shown that the gaps that we established (in Theorem 5.21 or 5.11) extend to gaps that are linear in s, on the sequences of matrices $n' \times n'$ that we construct. For example, if $s \sim \log(n')$, then we produce gaps that grow logarithmically in the order n' of the covariance matrix.

Our technical results establish the strong potential for masking to improve on the (scaled) linx bound. So the next logical step is to work on optimizing the mask in this context. Similar work was carried out successfully for the spectral bound (see (Anstreicher and Lee, 2004) and (Burer and Lee, 2007)), where nonconvexity and nondifferentiability were the main difficulties to overcome. In the context of the linx bound, even at smooth points, it is not easy to get a handle on the necessary derivative information. There is also the potential to incorporate the "mixing" technique of (Chen, Fampa, Lambert, and Lee, 2021) on top of mask optimization. We are currently working in this direction, and we plan to report on algorithmic results (with experimentation on benchmark data) for mask optimization in a future paper.

CHAPTER 6

On Algorithms for Mask Optimization for Anstreicher's linx Bound

6.1 Introduction

From Chapter 5, it is established that masking can significantly enhance the linx bound, independent of other methods. That chapter detailed the optimization of masking patterns for specific matrix classes, notably diagonal and 2-by-2 block diagonal matrices, illustrating substantial improvements in the linx bound proportional to the problem size n for many instances. This chapter steps further by introducing an advanced quasi-Newton algorithm designed to compute an improved mask over the baseline all-ones mask (original bound) for the linx bound. Due to the Hadamard product interaction between the covariance matrix C and mask M, the *masked* linx bound presents nonconvexity and nondifferentiability challenges, alongside large-scale optimization issues. To address these complexities, we utilized the Limited Memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm, which has demonstrated efficiency in our context, as supported by multiple references (Liu and Nocedal, 1989; Nash and Nocedal, 1991; Morales, 2002; Lewis and Overton, 2013; Berahas, Nocedal, and Takác, 2016). An interior point method is adopted to handle the positive-semidefinite constraint. This chapter presents a comprehensive exposition of the adapted L-BFGS algorithm, including hyperparameter configurations. Our numerical experiments validate the algorithm's capability to consistently identify more effective masks than the all-ones mask across various instances, thereby confirming its practical utility.

In Section 6.2, we investigate several key properties of the masking for the linx bound. This includes the confirmation of an optimal mask's existence, the nonconvex nature of the mask, and the analysis of its directional derivative. These insights lay the groundwork for the subsequent development of the L-BFGS algorithm. Section 6.3 transforms the masking optimization problem into a barrier problem to handle the positive-semidefinite constraint.

We also develop the specific formula of the directional derivative for the linx bound with respect to the mask. This formula is then used to develop the pseudo-gradient that is pivotal for updating the Hessian approximation within our tailored L-BFGS algorithm. This section also details each step of the algorithm and outlines the hyperparameter configuration. To address the critical impact of initialization diversity on algorithm performance, owing to the nonconvexity, we introduce four distinct mask initialization methods. In Section 6.4, the algorithm's efficacy is evidenced through extensive numerical experiments. Notably, our approach yields superior masks compared to the standard all-ones mask for many instances, particularly when the subset size s is small relative to n (smaller than 20 for $n = 63$, smaller than 25 for $n = 90$, and smaller than 40 for $n = 124$), even with optimal scaling applied. This enhanced masking capability, coupled with scaling and complementation (CMESP-comp), further elevates the linx bound for cases where s is relatively large compared to n (larger than 42 for $n = 63$). Additionally, our results demonstrate the algorithm's stability across various mask initializations. In §6.5, we present some concluding remarks.

6.2 Mask properties for the linx bound

The *masked* linx bound can be represented by replacing C with $C \circ M$ in linx. Our primary objective is to determine the optimal mask, which entails solving the corresponding problem:

$$z^*_{\text{mlinx}}(C, s) = \min \{ z_{\text{linx}}(C, s; M) \ : \ M \in \mathcal{M}_n \} . \tag{mopt}$$

Initially, we demonstrate that when $z(C, s) > 0$, the optimal value specified in mopt is attainable.

Proposition 6.1. *Provided that the optimal value of MESP is finite, there exists an* $M^* \in \mathcal{M}_n$ *for which the equality* $z^*_{mlinx}(C, s) = z_{linx}(C, s; M^*)$ *holds true.*

Proof. We first establish that for any continuous function f defined over a compact domain $\mathcal{S}$, which possesses a lower bound and is not infinity everywhere, there necessarily exists a point within $\mathcal{S}$ at which f achieves its minimum value. Specifically, there is a point $x^* \in \mathcal{S}$ satisfying the condition:

$$f(x^*) = \min_{x \in \mathcal{S}} f(x).$$

We prove by contradiction. Given that f is both lower bounded and not infinity everywhere, assume, contrarily, that no point $x \in \mathcal{S}$ exists such that $f(x)$ equals the minimum of f over $\mathcal{S}$. This assumption implies the existence of an infinite sequence $\{x_k\} \subset \mathcal{S}$, where $\{f(x_k)\}$

monotonically decreases and converges to the minimum of f over $\mathcal{S}$ as k approaches infinity. Due to the compactness of $\mathcal{S}$, the sequence $\{x_k\}$ contains a convergent subsequence $\{x_{k_i}\}$ converging to some $\hat{x} \in \mathcal{S}$. Since f is continuous, it follows that $f(\hat{x}) = f\left(\lim_{i\to\infty} x_{k_i}\right) = \lim_{i\to\infty} f(x_{k_i})$, which equals the minimum of f over $\mathcal{S}$, thus leading to a contradiction.

In light of the preceding discussion, our proof hinges on demonstrating that $\mathcal{M}_n$ is compact and that $z_{\text{linx}}(C, s; M)$ is continuous, possesses a lower bound, and is not infinity everywhere across $\mathcal{M}_n$.

The compactness of $\mathcal{M}_n$ is attributed to two key factors: firstly, the boundedness of the eigenvalues as per the Gershgorin circle theorem (Horn and Johnson, 1985, Theorem 6.1.1), and secondly, the continuity of eigenvalues relative to matrix elements, a concept detailed in (Horn and Johnson, 1985, Appendix D). Furthermore, the continuity of $z_{\text{linx}}(C, s; M)$ is established by Theorem 4.12. The lower boundedness of $z_{\text{linx}}(C, s; M)$ is inferred from the inequality $z_{\text{linx}}(C, s; M) \geq z(C, s)$, combined with the finiteness of $z(C, s)$. Lastly, the finiteness of the eigenvalues of the matrix within the $\text{ldet}(\cdot)$ function of $f_{\text{linx}}(x)$, and consequently, the non-infinite nature of $z_{\text{linx}}(C, s; M)$ across $\mathcal{M}_n$, are deduced from the Gershgorin circle theorem and the fact that $x \in [0, 1]^n$. □

Unfortunately, the masked linx bound (and, in fact, most upper bounds of MESP) is not always convex in the mask M even when the dimension of C is 2.

Proposition 6.2. *Given* $C := \begin{pmatrix} a & 1 \\ 1 & a \end{pmatrix}$ *where* $a \geq 1$ *and* $s := 1$, *and* $\mathcal{M}_2 = \left\{ \begin{pmatrix} 1 & m \\ m & 1 \end{pmatrix} : m^2 \leq 1 \right\}$. *When* a *is large enough,* $z_{linx}(C, s; M)$ *is not always convex in* $M \in \mathcal{M}_2$.

Proof. By (Chen, Fampa, and Lee, 2022, Theorem 2.10), we know

$$\begin{aligned} z_{\text{linx}}(C, s; M) &= \max_{x_1+x_2=1; x_1, x_2 \geq 0} \tfrac{1}{2} \log\left(\left(m^2 + 1 - a^2\right)^2 x_1 x_2 + (a x_1 + a x_2)^2 \right) \\ &= \tfrac{1}{2} \log\left(\tfrac{\left(m^2+1-a^2\right)^2}{4} + a^2 \right). \end{aligned}$$

It is straightforward to confirm that $z_{\text{linx}}(C, s; M)$ is convex in M if and only if $z_{\text{linx}}(C, s; M)$ is convex in m. We now compute the second derivative of $z_{\text{linx}}(C, s; M)$ with respect to m,

$$\begin{aligned} \frac{\partial z_{\text{linx}}(C, s; M)}{\partial m} &= \tfrac{1}{2} \frac{m\left(m^2+1-a^2\right)}{\left(m^2+1-a^2\right)^2/4+a^2}, \\ \frac{\partial^2 z_{\text{linx}}(C, s; M)}{\partial m^2} &= \tfrac{1}{8} \frac{\left(3m^2+1-a^2\right)\left(\left(m^2+1-a^2\right)^2+4a^2\right) - 4m^2\left(m^2+1-a^2\right)^2}{\left(\left(m^2+1-a^2\right)^2/4+a^2\right)^2}. \end{aligned} \tag{6.1}$$

Because $m^2 \leq 1$, when a is sufficiently large, the numerator of the right-hand side of (6.1) is dominated by the negative term $O(-a^6)$. This dominance implies that $\frac{\partial^2 z_{\text{linx}}(C,s;M)}{\partial m^2}$ is negative when a is large enough, and thus, $z_{\text{linx}}(C, s; M)$ is not always convex in m. □

Nonetheless, even in the absence of convexity, we can still employ strategies from nonconvex optimization to enhance the masked linx bound. This is because the masked linx bound is locally Lipschitz in the mask by Theorem 4.12, and thus differentiable almost everywhere by Rademacher's theorem, a direct application of Theorem 4.12.

Theorem 6.3. *For any $M \in \mathcal{M}_n$, the masked linx bound $z_{linx}(C, s; M)$ is locally Lipschitz and differentiable almost everywhere around M, and directly differentiable at M in any feasible direction ΔM with formula*

$$\partial z_{linx}(C, s; M; \Delta M) = \max_{x \in \mathcal{S}^*(C,s;M)} \left(\frac{\partial f_{linx}(C, s; M; x)}{\partial M} \right)^{\mathsf{T}} \Delta M$$

where $\mathcal{S}^*(C, s; M)$ is the set of optimal solutions of masked linx. Furthermore, if $\mathcal{S}^*(C, s; M)$ is a singleton, then $z_{linx}(C, s; M)$ is differentiable at M.

Theorem 2.16 provides a sufficient condition for $\mathcal{S}^*(C, s; M)$ to be a singleton. It can be easily inferred from there that $z_{\text{linx}}(C, s; M)$ is actually differentiable almost everywhere.

Corollary 6.4. *$z_{linx}(C, s; M)$ is not differentiable only on a zero-measure subset of $\mathcal{M}_n$.*

6.3 Algorithms for mask optimization for the linx bound

Leveraging the above properties, we can formulate effective algorithms for mopt. Notably, considering the constraint $M \in \mathcal{M}_n$ of positive-semidefiniteness, we employ the interior point method by addressing a barrier problem instead of tackling mopt directly. This is represented as:

$$z_{\text{mlinx}}(C, s; \mu) = \min \left\{ z_{\text{linx}}(C, s; M) - \mu \ln \det(M) : M \in \tilde{\mathcal{M}}_n \right\}. \qquad \text{(mopt-barrier)}$$

Here, $\mu > 0$ is the barrier parameter, and $\tilde{\mathcal{M}}_n$ denotes the set of symmetric matrices of order n with unit diagonals. As per established understanding (Nocedal and Wright, 2006), as μ increases, the optimal solution and optimal value of mopt-barrier asymptotically converge to those of mopt.

To circumvent the complexities associated with tensor notation in matrix derivatives, we adopt the vectorization technique described in (Anstreicher and Lee, 2004). This method transforms the off-diagonal elements of a matrix M in $\tilde{\mathcal{M}}_n$ into a vector $y \in \mathbb{R}^{\frac{n(n-1)}{2}}$ using the operation $y = \text{svec}(M) := (M_{21}, M_{31}, \ldots, M_{n1}, M_{32}, \ldots, M_{n(n-1)})^\mathsf{T}$. Specifically, the $n(j-1) - \frac{j(j-1)}{2} + i - j$th element of y is represented as y_{ij}, where $y_{ij} = M_{ij}$. Additionally, we define $\text{smat}(y)$ as the inverse of the vectorization process, satisfying $\text{smat}(y) := M$ when $y = \text{svec}(M)$. These notations facilitate the conversion of derivatives with respect to matrix M into derivatives with respect to vector y, simplifying the computation of second-order derivatives or its approximations.

The problem delineated in mopt-barrier may manifest as a nonconvex and nonsmooth optimization challenge, albeit with differentiability in almost all regions. Notably, the expression $\frac{n(n-1)}{2}$ exhibits a quadratic rate of increase relative to n, surpassing 10,000 when n reaches a mere 142. This observation categorizes mopt-barrier as a large-scale optimization problem. In such contexts, the Limited Memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm emerges as an good solution strategy. This method is reputed for its efficacy in handling large-scale, non-convex, and possibly nonsmooth problems, as supported by numerous studies (Liu and Nocedal, 1989; Nash and Nocedal, 1991; Morales, 2002; Lewis and Overton, 2013; Berahas, Nocedal, and Takác, 2016). L-BFGS specifically targets extensive problems where Hessian matrix computation is either infeasible or prohibitively expensive, or where such matrices are neither sparse nor readily computable. Distinct from conventional methods that necessitate a full Hessian approximation, L-BFGS conserves computational resources by storing a limited number (m, typically between 3 and 20, and 17 in our analysis) of vector pairs that implicitly represent the Hessian approximation (Nocedal and Wright, 2006). This strategy effectively reduces the computational complexity per iteration from $O(n^4)$ to $O(mn^2)$, with m being significantly smaller than n. We want to point out here that although some findings by (Asl and Overton, 2021) indicate potential shortcomings in the L-BFGS method. In contrast, our numerical experiments consistently demonstrate the superiority of L-BFGS over the full memory BFGS method, as evidenced by reduced computational time and memory requirements, alongside lower linx bounds. These results suggest that our experimental framework differs from that of (Asl and Overton, 2021), thus validating the selection of L-BFGS for our purposes.

In the implementation of the L-BFGS algorithm, the gradient approximation is necessitated for updating the Hessian approximation. According to Theorem 6.3, we can choose $x^* \in \mathcal{S}^*(C, s; M)$ and employ $\frac{\partial f_{\text{linx}}(C,s;M;x^*)}{\partial y}$ as the gradient approximation. Theorem 6.3 further asserts that, should $\mathcal{S}^*(C, s; M)$ be a singleton, $\frac{\partial f_{\text{linx}}(C,s;M;x^*)}{\partial y}$ coincides with the exact gradient. The subsequent proposition details the expression of $\frac{\partial f_{\text{linx}}(C,slM;x)}{\partial y_{ij}}$.

Proposition 6.5. *Given $M \in \mathcal{M}_n$, define $F(C,s;M;x) := (C \circ M)\,\mathrm{Diag}(x)(C \circ M) + \mathrm{Diag}(\mathbf{e} - x)$, $A(C,s;M;x) := F(C,s;M;x)^{-1}(C \circ M)$, and $B(C,s;M;x) := (C \circ M)F(C,s;M;x)^{-1}(C \circ M)$. For simplicity, we will denote $F(C,s;M;x)$, $A(C,s;M;x)$, and $B(C,s;M;x)$ as F, A, and B respectively when it is clear from the context. Denote $y = \mathrm{svec}(M)$, where $\mathrm{svec}(M) := (M_{21}, M_{31}, \ldots, M_{n1}, M_{32}, \ldots, M_{n(n-1)})^{\mathsf{T}}$. Specifically, the $n(j-1) - \frac{j(j-1)}{2} + i - j$th element of y is represented as y_{ij}, where $y_{ij} = M_{ij}$. Then for any $1 \le i < j \le n$, we have*

$$\frac{\partial f_{linx}(C,s;M;x)}{\partial y_{ij}} = C_{ij}\left(x_j A_{ij} + x_i A_{ji}\right).$$

Proof. Note that

$$\begin{aligned}
\frac{\partial f_{\mathrm{linx}}(C,s;M;x)}{\partial y_{ij}} &= \frac{\partial f_{\mathrm{linx}}(C,s;M;x)}{\partial M} \bullet \frac{\partial M}{\partial y_{ij}} \\
&= \frac{\partial f_{\mathrm{linx}}(C,s;M;x)}{\partial M} \bullet (E_{ij} + E_{ji}) \\
&= \frac{\partial f_{\mathrm{linx}}(C,s;M;x)}{\partial M_{ji}} + \frac{\partial f_{\mathrm{linx}}(C,s;M;x)}{\partial M_{ij}} \\
&= \tfrac{1}{2}F^{-1} \bullet \big(C_{ji}x_i E_{ji}(C \circ M) + (C \circ M)x_j C_{ji}E_{ji} + \\
&\quad C_{ij}x_j E_{ij}(C \circ M) + (C \circ M)x_i C_{ij}E_{ij}\big).
\end{aligned}$$

where E_{ij} is an order-n matrix with the ith row and jth column element equal one and other elements equal zero. We can further simplify

$$\begin{aligned}
F^{-1} \bullet (C_{ij}x_j E_{ij}(C \circ M)) &= C_{ij}x_j\,\mathrm{Trace}\left(F^{-1}\mathbf{e}_i\mathbf{e}_j^{\mathsf{T}}(C \circ M)\right) \\
&= C_{ij}x_j\,\mathrm{Trace}\left(\mathbf{e}_j^{\mathsf{T}}(C \circ M)F^{-1}\mathbf{e}_i\right) \\
&= C_{ij}x_j A_{ij}
\end{aligned}$$

where the equations comes from the symmetry of F and $C \circ M$. Similarly, we have

$$\begin{aligned}
F^{-1} \bullet ((C \circ M)x_i C_{ij}E_{ij}) &= C_{ij}x_i A_{ji} \\
F^{-1} \bullet (C_{ji}x_i E_{ji}(C \circ M)) &= C_{ji}x_i A_{ji} \\
F^{-1} \bullet ((C \circ M)x_j C_{ji}E_{ji}) &= C_{ji}x_j A_{ij}.
\end{aligned}$$

Aggregating all the above gives us the first-order derivative

$$\frac{\partial f_{\mathrm{linx}}(C,s;M;x)}{\partial y_{ij}} = C_{ij}\left(x_j A_{ij} + x_i A_{ji}\right).$$

□

According to the derivation of Proposition 6.5, we can also derive the formula of $\frac{\partial f_{\text{linx}}(C,s;M;x)}{\partial y}$.

Corollary 6.6. *Given $M \in \mathcal{M}_n$ and $y = \text{svec}(M)$, let F, A, B be that defined in Proposition 6.5. Then*

$$\frac{\partial f_{linx}(C,s;M;x)}{\partial y} = \text{svec}\left(C \circ \left(A\,\text{Diag}(x) + \text{Diag}(x)A^{\mathsf{T}}\right)\right).$$

We also present $\frac{\partial\, \text{ldet}(M)}{\partial y}$ here for completeness.

Proposition 6.7. *Given $M \in \mathcal{M}_n$ and $y = \text{svec}(M)$, let F, A, B be that defined in Proposition 6.5. Then*

$$\frac{\partial\, \text{ldet}\, M}{\partial y} = \text{svec}(M + M^{\mathsf{T}}).$$

Proof. Note that

$$\begin{aligned}\frac{\partial\, \text{ldet}\, M}{\partial y_{ij}} &= \frac{\partial\, \text{ldet}\, M}{\partial M} \bullet \frac{\partial M}{\partial y_{ij}} = \frac{\partial\, \text{ldet}\, M}{\partial M} \bullet (E_{ij} + E_{ji}) \\ &= \frac{\partial\, \text{ldet}\, M}{\partial M_{ji}} + \frac{\partial\, \text{ldet}\, M}{\partial M_{ij}} = \text{Trace}\,(M^{-1}(E_{ji} + E_{ij})) = M^{-1}_{ij} + M^{-1}_{ji}.\end{aligned}$$

□

In the following, we delineate a single iteration of the L-BFGS algorithm as applied to the barrier problem mopt-barrier, which is reiterated below for clarity:

$$z_{\text{mlinx}}(C, s; \mu) = \min\left\{z_{\text{linx}}(C, s; M) - \mu \ln\det(M) : M \in \tilde{\mathcal{M}}_n\right\}.$$

Adhering to the methodologies delineated in (Nocedal and Wright, 2006, Chapter 7.2), we use m to represent the number of historical vector pairs retained for computing the Hessian approximation in each L-BFGS iteration. Notably, during the initial k iterations, the L-BFGS algorithm operates analogously to the standard BFGS algorithm, as described in (Nocedal and Wright, 2006, Chapter 6).

For notation clarity, we denote $g_y(C, s; M; \mu) := \frac{\partial f_{\text{linx}}(C,s;M;x^*)}{\partial y} - \mu\frac{\partial\, \text{ldet}(M)}{\partial y}$ where $y = \text{svec}(M)$ and $x^* \in \mathcal{S}^*(C, s; M)$. At the kth iteration, we denote the mask as M_k, $y_k = \text{svec}(M_k)$, and $g_k := g_y(C, s; M_k; \mu)$ for simplicity. We further denote $s_k := y_{k+1} - y_k$, $t_k := g_{k+1} - g_k$, and $\rho_k := \frac{1}{s_k^{\mathsf{T}} t_k}$.

In the k^{th} iteration of the algorithm, the Hessian approximation H_k^0 is initialized as an

identity matrix scaled by the factor γ_k, computed as

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}.$$

This scaling factor γ_k aims to approximate the magnitude of the actual Hessian matrix in the direction of the latest search, as elucidated in (Nocedal and Wright, 2006, Chapter 6). Such a strategic selection of γ_k is instrumental in ensuring the search direction's appropriateness, typically allowing the algorithm to accept a unit step length $\alpha_k = 1$ in the majority of iterations. There could be some cases that the curvature information in the last iteration fails, i.e., $s_{k-1}^T y_{k-1}$ is too small (typically smaller than 10^{-6}), in which case, $\frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$ is not an appropriate scaling factor anymore, and we set $\gamma_k = \gamma_{k-1}$ instead ($\gamma_0 = 0$ in default).

In the L-BFGS framework, we update H_k^0 by collecting the last k vector pairs $\{s_i, y_i\}$, where $i = k - m, \ldots, k - 1$. The difference between L-BFGS and the traditional BFGS lies in the limited-memory feature intrinsic to L-BFGS. Contrary to the BFGS algorithm, which initially computes the Hessian approximation H_k and then its product with g_k, L-BFGS directly ascertains the search direction $r_k := -H_k g_k$ (see (Nocedal and Wright, 2006, Algorithm 7.4)), which reduces both memory and computational overhead, curtailing the complexity from $O(n^4)$ to $O(mn^2)$. Importantly, in our methodology, we omit any s_i, y_i pairs where the curvature information is insufficient, specifically when $s_{k-1}^T y_{k-1}$ falls below a threshold, usually set at 10^{-6}, as suggested in (Nocedal and Wright, 2006).

Upon computing the search direction r_k in the L-BFGS algorithm, we first evaluate the applicability of the unit step length, adhering to the standard protocol in Newton-type methods. If feasible, this step length is adopted. Otherwise, the procedure engages the weak-Wolfe line search as delineated in (Nocedal and Wright, 2006, Algorithm 3.5), to determine a step length α_k. This step length ensures the positive-definiteness of $M_k + \alpha_k R_k$, with $R_k := \mathrm{smat}(r_k) - I$ representing the matrix format of r_k. The function $\phi_k(\alpha_k) := f_{\mathrm{linx}}(C, s; M_k + \alpha_k R_k) - \mu \mathrm{ldet}(M_k + \alpha_k R_k)$ is formulated correspondingly. The weak Wolfe conditions are subsequently formalized as follows:

$$\phi_k(\alpha_k) \leq \phi_k(0) + c_1 g_k^\mathsf{T} r_k, \tag{6.2}$$

$$g_y(C, s; M_k + \alpha_k R; \mu)^\mathsf{T} r_k \geq c_2 g_k^\mathsf{T} r_k, \tag{6.3}$$

where c_1 and c_2 are parameters for the weak-Wolfe conditions, typically set to 10^{-6} and 0.95, respectively. The line search concludes upon satisfying these conditions or when the search interval contracts below a certain threshold, e.g., 10^{-2}. In the latter case, we adopt the midpoint of the final interval as the step length. This approach, often necessitated by the potential non-convexity of mopt-barrier, facilitates progression beyond local minima, thus

increasing the likelihood of finding a superior mask. We also employ y^* to record the best mask encountered during the running of the algorithm.

In employing the interior point method, efficiency is contingent upon defining the update rules for the barrier parameter μ and the termination criteria for the inner iteration in solving mopt-barrier. We set the initial barrier parameter $\mu_1 = 10^{-1}$ and update $\mu_j = 10^{-1}\mu_{j-1}$ following each outer iteration until $\mu_{10} = 10^{-10}$. The inner iteration is terminated under any of the following conditions:

1. The norm of the approximate gradient is below a threshold, 10^{-8};
2. The number of inner iterations reaches a predetermined limit, 20 iterations;
3. A certain number of consecutive inner iterations, which is set to 20, fail to achieve an objective value decrease greater than 10^{-8}.

Finally, the potential nonconvex nature of the optimization problem as outlined in mopt necessitates the selection of an effective initial condition. This strategy is pivotal for achieving expedited convergence and minimizing the final mask values. The most naive way to do initialization is to generate an m-by-n matrix A, populated with elements that are independent Gaussian random variables, each with zero mean and unit variance. A diagonal matrix D is then defined, with diagonal elements being the square roots of the inverse of the diagonal elements in A. The mask initialization is computed as $M = SAS$. A notable limitation of this method is the tendency for many off-diagonal elements to approach zero when m and n are large, diminishing the diversity in mask initialization and potentially impacting the effectiveness of mask optimization under nonconvexity. The following three alternative methods address this issue from diverse perspectives, yielding random correlation matrices that uniformly populate the space $\mathcal{M}_n^o := \{X : X \in \mathcal{M}_n, X \succ 0\}$.

1. `randcorr_exOnion(m, n)`: (Ghosh and Henderson, 2003) introduced a technique for constructing a correlation matrix, commencing with a unidimensional matrix and progressively expanding it by adding dimensions sequentially. This approach was further refined by Lewandowski (2009), who termed it the "extended onion method", designed to generate random correlation matrices that uniformly occupy the specified space $\mathcal{M}_n^o := X : X \in \mathcal{M}_n, X \succ 0$.
2. `randcorr_vine(m, n)`: (Joe, 2006) proposed a method for parameterizing positive-definite correlation matrices using correlations and partial correlations. By independently sampling these (partial) correlations from a beta distribution, they achieved uniform generation of random correlation matrices over $\mathcal{M}_n^o$. Notably, this approach

incorporates the use of D-vine, a graphical model for delineating dependence structures in high-dimensional probability distributions, as detailed in (Lewandowski, Kurowicka, and Joe, 2009).

3. `randcorr_Cholesky(m, n)`: Pourahmadi et al. (Pourahmadi and Wang, 2015) developed an algorithm to generate random correlation matrices, utilizing Cholesky factorization and $\frac{n(n-1)}{2}$ hyperspherical coordinates. This method involves sampling angles from a specific distribution and subsequently converting them into standard correlation matrix form.

4. `randcorr_MatlabGallery(m, n)`: Matlab gallery toolbox implemented an algorithm to construct correlation matrices, with the option to specify predetermined eigenvalues. This methodology is rooted in the algorithm proposed by (Bendel and Mickey, 1978) and later refined by (Davies and Higham, 2000). It entails modifying a matrix to have specified eigenvalues (randomly generated if not specified) using a series of Givens rotations, resulting in a diagonal matrix of ones. Our rationale for selecting this method is its ability to produce correlation matrices with a varied eigenvalue pattern.

Last but not least, we will employ the advantages of scaling identified in (Anstreicher, 2020), supplemented by the findings of (Chen, Fampa, Lambert, and Lee, 2021; Chen, Fampa, and Lee, 2022). The research demonstrates that appropriate scaling parameters markedly enhance the linx bound, a trend persisting despite masking interventions (Chen, Fampa, and Lee, 2022). Notably, scaling and masking effects are distinct and non-overlapping techniques for improving linx bound. Our approach involves initially determining the optimal scaling parameter γ for linx in the absence of masking. This parameter γ is then integrated into the constant C, modifying it to $\sqrt{\gamma}C$. Subsequently, we apply the proposed algorithm to resolve mopt, holding γ constant. Upon determining the final mask M, it is incorporated into C, reformulating it as $C \circ M$. Finally, we recalibrate the optimal scaling parameter under this definitive mask, thereby establishing the final masked linx bound.

6.4 Experiments

In our research, we assessed the efficacy of the L-BFGS algorithm across various parameters, applied to benchmark datasets. These evaluations employed three established covariance matrices of sizes $n = 63, 90, 124$ (as detailed in (Anstreicher, 2018, 2020; Anstreicher, Fampa, Lee, and Williams, 1999; Ko, Lee, and Queyranne, 1995; Lee, 1998)). For each matrix dimension n, signifying a distinct benchmark covariance matrix, we explored a range of s

values, creating diverse MESP and CMESP cases. We repeated the experiments for each instance five times and record the best results. The experiments were executed on an Intel Xeon E5-2667 v4 @ 3.20 GHz system with Windows OS, featuring 8 physical cores (16 virtual cores) and 128 GB RAM.

In the context of masked linx bounds, the all-ones mask J, replicates the unmasked linx bound. This mask will serve as the fundamental baseline in our analysis and the integrality gap (the difference between the masked linx bound and the best lower bound we can obtain) associated with J (Gap-J) will be computed for comparative analysis. For each instance, we compute the integrality gap (Gap) to assess the found mask's quality.

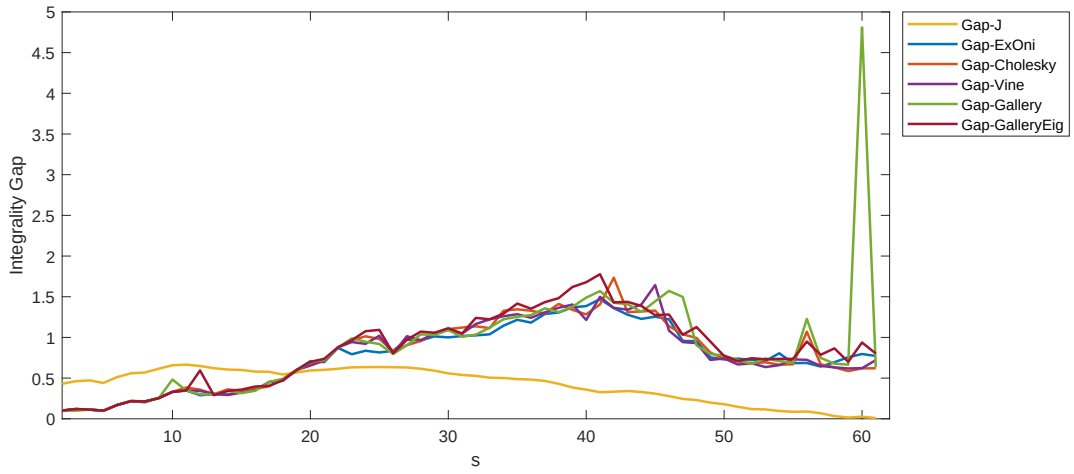

Figure 6.1: Integrality gaps for un-masked linx bound (Gap-J) and masked linx bound with extended onion, vine, random Cholesky, Matlab gallery, and Matlab gallery with specified eigenvalues initialization respectively.

The initial phase of our study involved assessing the impact of various initializations on our algorithm's performance, as depicted in Figure 6.1. Furthermore, we observed that the unit step length was employed most of the time. These results demonstrated uniform integrality gaps across different initializations, underscoring the algorithm's robustness. Consequently, the extended onion initialization was selected as the standard for subsequent analyses. It should be noted that our initialization strategy does not involve an all-ones mask, despite its dominance over the mask in our algorithm's outputs for some instances. This is attributed to our utilization of the interior point method for managing positive-semidefinite constraints. Initiation at the boundary of the positive-semidefinite cone is deliberately avoided, as it would impede algorithmic progress in high-dimensional contexts, primarily due to the search direction frequently pointing outside the positive-semidefinite cone.

Additionally, it was observed that, in scenarios where the subset size s is relatively smaller

than the data size n (smaller than 20 for $n = 63$, smaller than 25 for $n = 90$, and smaller than 40 for $n = 124$), our algorithm could generate masks that yield superior upper bounds compared to unmasked scenarios. However, for relatively large subset sizes, the algorithm failed to produce a mask that outperforms the all-ones mask. An analysis of the algorithm's operational dynamics revealed that termination typically occurred as the mask iterate approached the positive-semidefinite cone's boundary. This proximity hindered further algorithmic progression due to its intrinsic interior properties.

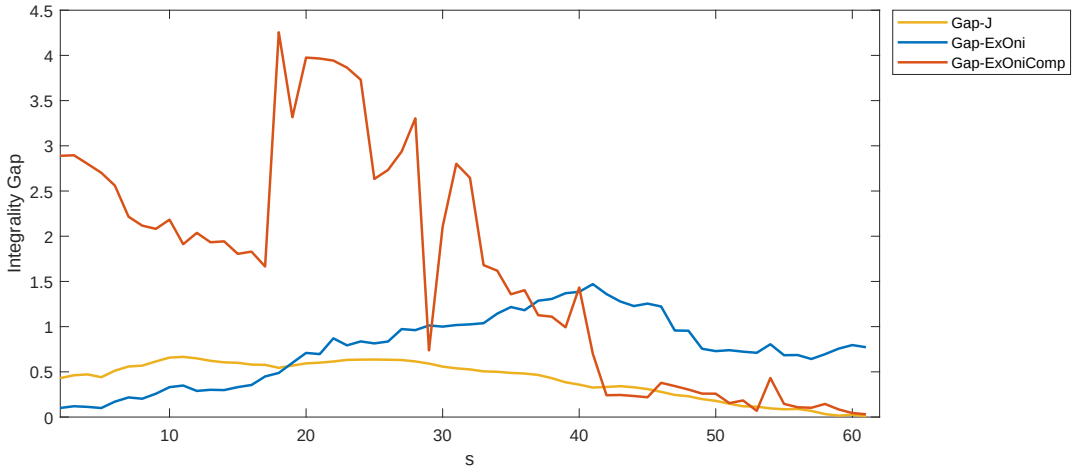

Figure 6.2: Integrality gaps for un-masked linx bound (Gap-J), masked linx bound with extended onion initialization, and complemented masked linx bound with extended onion initialization.

Building on the efficacy demonstrated by our algorithm for relatively small subset sizes s, we explored its application to the complementary linx bound. Notably, despite the established invariance of the linx bound under complementation as outlined in (Anstreicher, 2020), this property does not extend to scenarios involving masking. Our findings indicate that, with the application of our algorithm on the complementary linx bound, masks can be identified that yield similar or improved masked linx bounds for relatively large s values (larger than 42 for $n = 63$), as evidenced in Figures 6.2, 6.3, and 6.4. However, for intermediate subset sizes, neither the masked linx bound nor its complementary counterpart surpasses the unmasked linx bound in terms of achieving superior upper bounds.

6.5 Concluding remarks

Our findings indicate that the application of masking can significantly enhance the linx bound for a broad range of benchmark instances, particularly when the subset size s is relatively

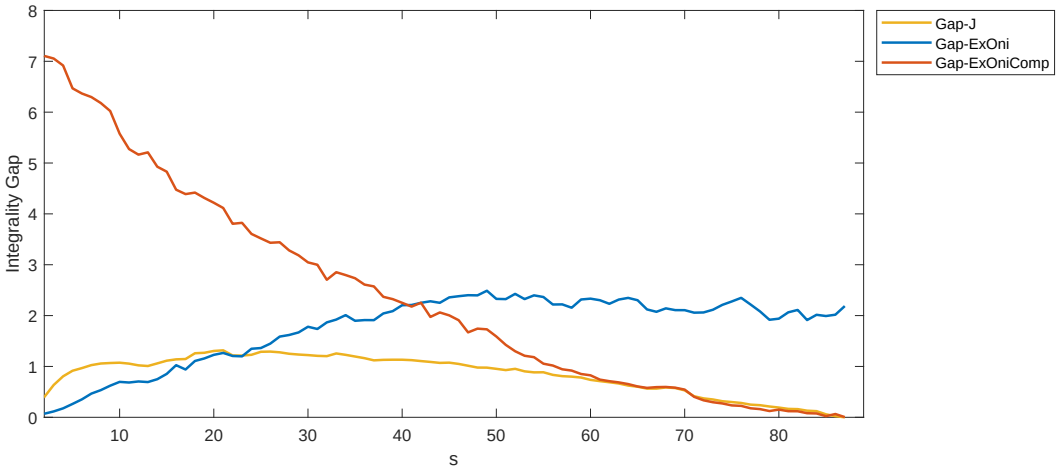

Figure 6.3: Integrality gaps for un-masked linx bound (Gap-J), masked linx bound with extended onion initialization, and complemented masked linx bound with extended onion initialization.

small in comparison to the overall problem size n. This improvement could be particularly beneficial in a branch-and-bound framework, especially in branches where these conditions prevail.

We have also developed a comprehensive L-BFGS algorithm to compute an effective mask for general problem instances. While this algorithm is specifically designed for the linx bound, its conceptual framework is versatile and can be readily adapted to other upper bounds in MESP, such as BQP, NLP, and Fact bounds. Extending this approach to these bounds can be a subject for future work.

Finally, it is observed that the interior point method might face limitations in facilitating further progress, especially at iterations near the boundary of the positive-semidefinite cone. Thus, there is room for enhancing the algorithm by permitting iterations at, or even beyond, the boundary, followed by projecting back. This adjustment may allow the use of an all-ones mask as the initial point, potentially leading to consistent improvements in the upper bounds over the original (all-ones mask) bound for all instances.

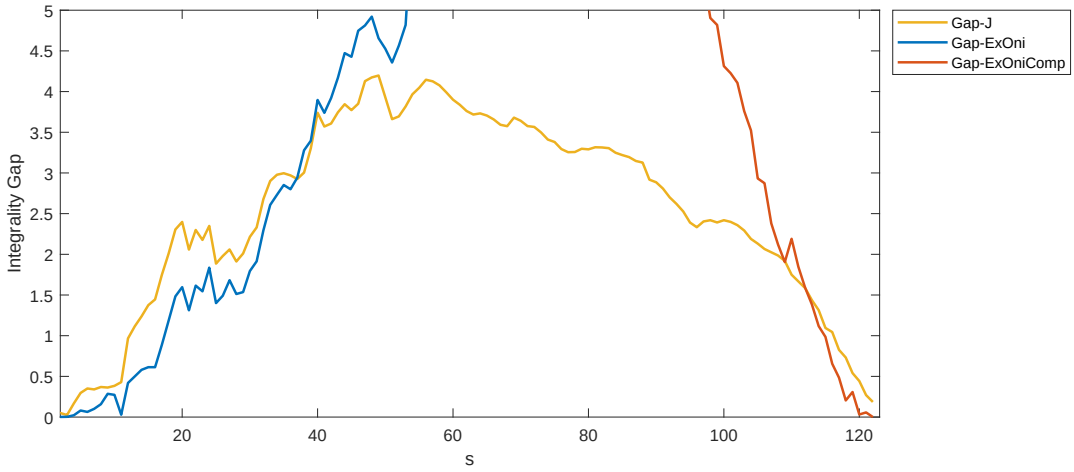

Figure 6.4: Integrality gaps for un-masked linx bound (Gap-J), masked linx bound with extended onion initialization, and complemented masked linx bound with extended onion initialization. *For readability, we truncate the y-axis range to* $[0, 5]$ *because we only care about relatively small and large* s.

BIBLIOGRAPHY

Hessa Al-Thani and Jon Lee. An R package for generating covariance matrices for maximum-entropy sampling from precipitation chemistry data. *SN Operations Research Forum*, Volume 1:Article 17 (21 pages), 2020a. https://doi.org/10.1007/s43069-020-0011-z.

Hessa Al-Thani and Jon Lee. `MESgenCov`, 2020b. https://github.com/hessakh/MESgenCov.

Hessa Al-Thani and Jon Lee. Tridiagonal maximum-entropy sampling and tridiagonal masks. *LAGOS 2021 proceedings, Procedia Computer Science*, 195:127–134, 2021.

Hessa Al-Thani and Jon Lee. Tridiagonal maximum-entropy sampling and tridiagonal masks. *Discrete Applied Mathematics*, 337:120–138, 2023.

Kurt M. Anstreicher. Maximum-entropy sampling and the Boolean quadric polytope. *Journal of Global Optimization*, 72(4):603–618, 2018.

Kurt M. Anstreicher. Efficient solution of maximum-entropy sampling problems. *Operations Research*, 68(6):1826–1835, 2020.

Kurt M. Anstreicher and Jon Lee. A masked spectral bound for maximum-entropy sampling. In *mODa 7—Advances in Model-Oriented Design and Analysis*, Contrib. Statist., pages 1–12. Physica, Heidelberg, 2004.

Kurt M. Anstreicher, Marcia Fampa, Jon Lee, and Joy Williams. Continuous relaxations for constrained maximum-entropy sampling. In *Integer Programming and Combinatorial Optimization (Vancouver, BC, 1996)*, volume 1084 of *Lecture Notes in Computer Science*, pages 234–248. Springer, Berlin, 1996.

Kurt M. Anstreicher, Marcia Fampa, Jon Lee, and Joy Williams. Using continuous nonlinear relaxations to solve constrained maximum-entropy sampling problems. *Mathematical Programming, Series A*, 85(2):221–240, 1999.

Kurt M. Anstreicher, Marcia Fampa, Jon Lee, and Joy Williams. Maximum-entropy remote sampling. *Discrete Applied Mathematics*, 108(3):211–226, 2001.

Azam Asl and Michael L. Overton. Behavior of limited memory bfgs when applied to nonsmooth functions and their nesterov smoothings. In *Numerical Analysis and Optimization: NAO-V, Muscat, Oman, January 2020 V*, pages 25–55. Springer, 2021.

Shrey Bagroy, Ponnurangam Kumaraguru, and Munmun De Choudhury. A social media based index of mental well-being in college campuses. In *Proceedings of the 2017 CHI Conference on Human factors in Computing Systems*, pages 1634–1646, 2017.

Mihály Bakonyi and Hugo J. Woerdeman. *Matrix completions, moments, and sums of Hermitian squares*. Princeton University Press, Princeton, NJ, 2011. ISBN 978-0-691-12889-4.

Sankar Basu, Charles A. Micchelli, and Peter Olsen. Maximum entropy and maximum likelihood criteria for feature selection from multivariate data. In *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 3, pages 267–270. IEEE, 2000.

Robert B. Bendel and M. Ray Mickey. Population correlation matrices for sampling experiments. *Communications in Statistics-Simulation and Computation*, 7(2):163–182, 1978.

Anil K. Bera and Sung Y. Park. Optimal portfolio diversification using the maximum entropy principle. *Econometric Reviews*, 27(4–6):484–512, 2008.

Albert S. Berahas, Jorge Nocedal, and Martin Takác. A multi-batch L-BFGS method for machine learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Alain Billionnet, Sourour Elloumi, Amélie Lambert, and Angelika Wiegele. Using a Conic Bundle method to accelerate both phases of a Quadratic Convex Reformulation. *INFORMS Journal on Computing*, 29:318–331, 2017.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Samuel Burer and Jon Lee. Solving maximum-entropy sampling problems using factored masks. *Mathematical Programming, Series B*, 109(2–3):263–281, 2007.

Zhongzhu Chen, Marcia Fampa, Amélie Lambert, and Jon Lee. Mixing convex-optimization bounds for maximum-entropy sampling. *Mathematical Programming, Series B*, 188:539–568, 2021.

Zhongzhu Chen, Marcia Fampa, and Jon Lee. Masking Anstreicher's linx bound for improved entropy bounds. *Operations Research*, 2022.

Zhongzhu Chen, Marcia Fampa, and Jon Lee. On computing with some convex relaxations for the maximum-entropy sampling problem. *INFORMS Journal on Computing*, 35(2): 368–385, 2023.

Philip I. Davies and Nicholas J. Higham. Numerically stable generation of correlation matrices and their factors. *BIT Numerical Mathematics*, 40:640–651, 2000.

Santanu S Dey, Rahul Mazumder, and Guanyi Wang. Using ℓ_1-relaxation and integer programming to obtain dual bounds for sparse PCA. *Operations Research*, 70(3):1914–1932, 2022.

Marcia Fampa and Jon Lee. *Maximum-Entropy Sampling: Algorithms and Application*. Springer International Publishing, 2022. URL https://doi.org/10.1007/978-3-031-13078-6.

Valerii Fedorov and Jon Lee. Design of experiments in statistics. In *Handbook of semidefinite programming*, volume 27 of *Internat. Ser. Oper. Res. Management Sci.*, pages 511–532. Kluwer Acad. Publ., Boston, MA, 2000.

Anthony V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*, volume 165 of *Mathematics in Science and Engineering*. Academic Press, Inc., Orlando, FL, 1983. ISBN 0-12-254450-1.

Anthony V. Fiacco and Yo Ishizuka. Sensitivity and stability analysis for nonlinear programming. *Annals of Operations Research*, 27(1-4):215–235, 1990a.

Anthony V. Fiacco and Yo Ishizuka. Sensitivity and stability analysis for nonlinear programming. *Annals of Operations Research*, 27(1):215–235, 1990b.

Anthony V. Fiacco and Garth P. McCormick. *Nonlinear Programming : Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York, NY, USA, 1968. Reprint : Volume 4 of *SIAM Classics in Applied Mathematics*, SIAM Publications, Philadelphia, PA 19104–2688, USA, 1990.

Ilse Fischer, Gerald Gruber, Franz Rendl, and Renata Sotirov. Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-cut and equipartition. *Mathematical Programming*, 105:451–469, 2006.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1–2):95–110, 1956.

Soumyadip Ghosh and Shane G. Henderson. Behavior of the NORTA method for correlated random vector generation as the dimension increases. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(3):276–294, 2003.

Peter Guttorp, Nhu D. Le, Paul D. Sampson, and James V. Zidek. Using entropy in the redesign of an environmental monitoring network. In G.P. Patil, C.R. Rao, and N.P. Ross, editors, *Multivariate Environmental Statistics*, volume 6, pages 175–202. North-Holland, 1993.

Christoph Helmberg. The ConicBundle Library for Convex Optimization. https://www-user.tu-chemnitz.de/~helmberg/ConicBundle/, 2005–2019.

Jeffrey C. Hoch, Mark W. Maciejewski, Mehdi Mobli, Adam D. Schuyler, and Alan S. Stern. Nonuniform sampling and maximum entropy reconstruction in multidimensional nmr. *Accounts of Chemical Research*, 47(2):708–717, 2014.

Alan Hoffman, Jon Lee, and Joy Williams. New upper bounds for maximum-entropy sampling. In *mODa 6—Advances in Model-Oriented Design and Analysis (Puchberg/Schneeberg, 2001)*, Contrib. Statist., pages 143–153. Physica, Heidelberg, 2001.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, First edition, 1985. ISBN 0-521-38632-2.

Robert E. Hoskisson, Michael A. Hitt, Richard A. Johnson, and Douglas D. Moesel. Construct validity of an objective (entropy) categorical measure of diversification strategy. *Strategic Management Journal*, 14(3):215–235, 1993.

P. Jana, T.K. Roy, and S.K. Mazumder. Multi-objective mean-variance-skewness model for portfolio optimization. *Advanced Modeling and Optimization*, 9(1):181–193, 2007.

Xin Jin, Bamshad Mobasher, and Yanzan Zhou. A web recommendation system based on maximum entropy. In *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*, volume 1, pages 213–218. IEEE, 2005.

Harry Joe. Generating random correlation matrices based on partial correlations. *Journal of Multivariate Analysis*, 97(10):2177–2189, 2006.

Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum-entropy sampling. *Operations Research*, 43(4):684–691, 1995.

Jon Lee. Constrained maximum-entropy sampling. *Operations Research*, 46(5):655–664, 1998.

Jon Lee. *Encyclopedia of Environmetrics, A.H. El-Shaarawi and W.W. Piegorsch, eds.*, chapter Maximum entropy sampling, 2nd edition, pages 1570–1574. Wiley, 2012.

Jon Lee and Joy Williams. A linear integer programming bound for maximum-entropy sampling. *Mathematical Programming, Series B*, 94(2–3):247–256, 2003.

Jon Lee, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4):2053–2078, 2009/10.

Daniel Lewandowski, Dorota Kurowicka, and Harry Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9):1989–2001, 2009.

Adrian S. Lewis and Michael L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141:135–163, 2013.

Yongchun Li and Weijun Xie. Best principal submatrix selection for the maximum entropy sampling problem: Scalable algorithms and performance guarantees. *Operations Research*, 2023. https://doi.org/10.1287/opre.2023.2488.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

Johan Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, pages 284–289. IEEE, 2004.

José Luis Morales. A numerical study of limited memory BFGS methods. *Applied Mathematics Letters*, 15(4):481–487, 2002.

Jean-Jacques Moreau. Fonctionnelles convexes. *Séminaire Jean Leray*, 2:1–108, 1966.

NADP. National Acidic Deposition Program, National Trends Network. https://nadp.slh.wisc.edu/ntn/, 2018.

Stephen G. Nash and Jorge Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3):358–372, 1991.

Aleksandar Nikolov. Randomized rounding for the largest simplex problem. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 861–870, 2015.

Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

Daisuke Oyama and Tomoyuki Takenawa. On the (non-)differentiability of the optimal value function when the optimal solution is unique. *Journal of Mathematical Economics*, 76: 21–32, 2018.

Mohsen Pourahmadi and Xiao Wang. Distribution of random correlation matrices: Hyperspherical parameterization of the Cholesky factor. *Statistics & Probability Letters*, 106: 5–12, 2015.

Michael J.D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In *Nonlinear programming (Proc. Sympos., New York, 1975)*, pages 53–72. SIAM–AMS Proc., Vol. IX, 1976.

Lijing Qin and Xiaoyan Zhu. Promoting diversity in recommendation by entropy regularizer. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Zhicong Qiu, David J Miller, and George Kesidis. A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):917–933, 2016.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, 1997.

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

Michael C. Shewry and Henry P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 46:165–170, 1987.

Yuedong Song and Pietro Liò. A new approach for epileptic seizure detection: sample entropy based feature extraction and extreme learning machine. *Journal of Biomedical Science and Engineering*, 3(06):556, 2010.

Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü. SDPT3: A Matlab software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1-4):545–581, 1999.

Nam-Kiu Tsing, Michael K.H. Fan, and Erik I. Verriest. On analyticity of functions involving eigenvalues. *Linear Algebra and its Applications*, 207:159 – 180, 1994.

Constantin Zalinescu. *Convex Analysis in General Vector Spaces*. World Scientific, 2002.

Fuzhen Zhang, editor. *The Schur complement and its applications*, volume 4 of *Numerical Methods and Algorithms*. Springer-Verlag, New York, 2005.

James V. Zidek, Weimin Sun, and Nhu D. Le. Designing and integrating composite networks for monitoring multivariate Gaussian pollution fields. *Journal of the Royal Statistical Society. Series C. Applied Statistics*, 49(1):63–79, 2000.

Julian Zilly, Joachim M Buhmann, and Dwarikanath Mahapatra. Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation. *Computerized Medical Imaging and Graphics*, 55:28–41, 2017.