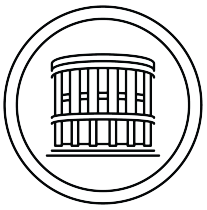
COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS



A CONIC OPTIMIZATION APPROACH FOR SOLVING MATRIX APPROXIMATION PROBLEMS

DISSERTATION THESIS

2023 Mgr. Terézia FULOVÁ

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

A CONIC OPTIMIZATION APPROACH FOR SOLVING MATRIX APPROXIMATION PROBLEMS

DISSERTATION THESIS

Study program:	Applied Mathematics
Study field:	Mathematics
Department:	Department of Applied Mathematics and Statistics
Supervisor:	doc. RNDr. Mária Trnovská, PhD.

2023 **Mgr. Terézia FULOVÁ**

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Mgr. Terézia Fulová
Study programme:	Applied Mathematics (Single degree study, Ph.D. III. deg., full time form)
Field of Study:	Mathematics
Type of Thesis:	Dissertation thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: A conic optimization approach for solving matrix approximation problems

Annotation: As a result of the interior-point methods development, the conical structure has become particularly desirable for some types of cones, thanks to the efficiency of the proposed algorithms. Conic problems can also be considered as an approximation tool for solving many non-convex problems. Exploring the possibilities for conic relaxation, the dual properties and the applications of conic programming in various fields is an interesting area of research. The dissertation thesis investigates matrix approximation problems in a unified framework, proposes a general conic optimization approach for their solution and demonstrates the results on specific sub-classes and various applications.

Tutor:	doc. RNDr. Mária Trnovská, PhD.
Department:	FMFI.KAMŠ - Department of Applied Mathematics and Statistics
Head of department:	doc. Mgr. Igor Melicherčík, PhD.

Assigned: 24.01.2019

Approved: 24.02.2021 prof. RNDr. Daniel Ševčovič, DrSc.
Guarantor of Study Programme

.....
Student

.....
Tutor

15708941

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Mgr. Terézia Fulová
Študijný program: aplikovaná matematika (Jednoodborové štúdium, doktorandské III. st., denná forma)
Študijný odbor: matematika
Typ záverečnej práce: dizertačná
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: A conic optimization approach for solving matrix approximation problems
Kónický optimalizačný prístup na riešenie problémov aproximácie matíc

Anotácia: V dôsledku rozvoja metód vnútorného bodu sa kónická štruktúra stala žiadanou najmä pre niektoré typy kužeľov, a to vďaka efektívnosti navrhnutých algoritmov. Kónické úlohy možno tiež považovať za aproximačný nástroj na riešene mnohých nekonvexných úloh. Skúmanie možností relaxácie pomocou kónických úloh, duálnych vlastností a aplikácie kónického programovania v rôznych odvetviach je zaujímavou oblasťou výskumu. Dizertačná práca skúma maticové aproximačné problémy v jednotnom rámci, navrhuje všeobecný prístup na ich riešenie pomocou kónickej optimalizácie a demonštruje výsledky na konkrétnych podtriedach a rôznych aplikáciách.

Školiteľ: doc. RNDr. Mária Trnovská, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: doc. Mgr. Igor Melicherčík, PhD.

Dátum zadania: 24.01.2019

Dátum schválenia: 24.02.2021 prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

.....
študent

.....
školiteľ

Acknowledgements I would like to express my sincere gratitude to my supervisor doc. RNDr. Mária Trnovská, PhD. for her guidance, assistance, and encouragement. Without her invaluable insights and constructive feedback, this work would not have been possible.

I would like to extend my gratitude to my family for their unconditional support throughout my academic journey. I am also grateful to my friends for being there to cheer me on and keep me motivated.

Last but not least, I would like to express my heartfelt thanks to my fiance for his boundless love and unwavering support. His belief in my abilities has been a constant source of motivation for me to persevere and complete this dissertation thesis.

Abstract

FULOVÁ, Terézia: A conic optimization approach for solving matrix approximation problems [Dissertation Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: doc. RNDr. Mária Trnovská, PhD., Bratislava, 2023, 150 p.

Matrix approximation problems are a subclass of constrained norm minimization problems. In this thesis, we handle a generalized formulation of matrix approximation problems to cover also well-known Procrustes problems. In general, matrix approximation problems have not been analyzed in a unified framework. The existing methods, which are designed to solve particular subclasses with a special structure of the feasibility set and a specific matrix norm in the objective. We aim to show that matrix approximation problems can be cast as conic programs with possible rank constraints. Therefore, we analyze various methods for solving rank-constrained optimization problems and propose a new solution algorithm based on using modified existing methods. Specifically, we address the problem of finding the nearest low-rank correlation matrix as well as different types of Procrustes problems, such as orthogonal, oblique, and semidefinite cases. We introduce a conic reformulation and demonstrate the correctness of this approach and the performance of our algorithm in numerous numerical experiments simulating real-life problems.

Keywords: conic optimization, matrix approximation problems, rank-constrained optimization problems, nearest correlation matrix, Procrustes problems

Abstrakt v štátnom jazyku

FULOVÁ, Terézia: Kónický optimalizačný prístup na riešenie problémov aproximácie matíc [Dizertačná práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. RNDr. Mária Trnovská, PhD., Bratislava, 2023, 150 s.

Úlohy aproximácie matíc tvoria podtriedu ohraničených úloh minimalizácie normy. V práci sa zaoberáme zovšeobecnenou formuláciou úloh aproximácie matíc s cieľom pokryť aj známe Procrustove úlohy. Vo všeobecnosti úlohy aproximácie matíc zatiaľ neboli analyzované v jednotnom rámci. Existujúce metódy na ich riešenie boli navrhnuté pre špeciálne podtriedy, kde má množina prípustných riešení špeciálnu štruktúru a účelová funkcia je definovaná pomocou konkrétnej maticovej normy. Naším cieľom je ukázať, že úlohy aproximácie matíc sa dajú naformulovať a riešiť ako kónické úlohy s prípadným ohraničením na hodnosť. Preto analyzujeme aj niekoľko metód na riešenie optimalizačných úloh s ohraničením na hodnosť s cieľom modifikovať ich a navrhnúť nový algoritmus na ich riešenie. Podrobnejšie sa venujeme úlohe hľadania najbližšej korelačnej matice nízkej hodnosti a niekoľkým typom Procrustových úloh, vrátane ortogonálnych, šikmých, a semidefinitných. Uvádzame ich kónickú reformuláciu a demonštrujeme správnosť tohto prístupu a správanie navrhnutého algoritmu pri riešení praktických úloh.

Kľúčové slová: kónická optimalizácia, úlohy aproximácie matíc, optimalizačné úlohy s ohraničením na hodnosť, najbližšia korelačná matica, Procrustove úlohy

Contents

List of Figures

List of Tables

List of symbols

$\mathbb{R}$	set of all real numbers
$\mathbb{N}_+$	set of all positive integers
$\mathbb{R}^n$	set of all n-dimensional vectors
$\mathbb{R}^n_+$	nonnegative orthant
$\mathbb{R}^{m\times n}$	set of all real $m \times n$ matrices
$\mathbb{S}^n$	set of all $n \times n$ symmetric matrices
$\mathbb{S}^n_+$	cone of all $n \times n$ symmetric positive semidefinite matrices
I_n	identity matrix $n \times n$
$\mathbf{1}_n$	unit n-dimensional column vector, i.e. $\mathbf{1}_n = (1, 1, ..., 1)^T$
$X \circ Y$	Hadamard product of X and Y, defined in Definition A.4
$rank(X)$	rank of matrix X
ε-$rank(X)$	numerical rank of matrix X, defined in Definition 3.1
$tr(X)$	trace of matrix X
$diag(X)$	column vector of diagonal entries of X
$Diag(x)$	diagonal matrix having vector x on diagonal
$svec(X)$	symmetric vectorization of symmetric matrix X defined as (14)
$\min\{...\}$	minimum of elements in braces
X^{-1}	inverse of a square matrix X satisfying $XX^{-1} = I$
$x^T y$	scalar product of vectors x and y
$\langle X, Y\rangle$	scalar product of symmetric X and Y, defined as $\langle X, Y\rangle = tr(XY)$
$\lfloor x \rfloor$	floor integer of a real number x
$\lceil x \rceil$	ceiling integer of a real number x
λ_i	i-th eigenvalue of a matrix
Λ	diagonal matrix with eigenvalues of a matrix on its diagonal
λ_{max}	the largest eigenvalue of X
σ_{max}	the largest singular value of X

$\|x\|$	absolute value of a number x
$\|x\|_1$	l_1 norm of a vector x, $\|x\|_1 = \sum_{i=1}^{n} \|x_i\|$
$\|X\|_F$	Frobenius norm of a matrix X, defined in Table 3
$\|X\|_1$	l_1 norm of a matrix X, defined in Table 3
$\|X\|_2$	spectral (l_2) norm of a matrix X, defined in Table 3
$\|X\|_\infty$	l_∞ norm of a matrix X, defined in Table 3
$X \succeq 0$	X is positive semidefinite, defined in Definition A.1
$X \succeq Y$	Löwner partial ordering, i.e., $X \succeq Y \Leftrightarrow X - Y \succeq 0$
$\mathcal{C}$	convex set
$\mathcal{K}$	convex cone
$\mathcal{L}$	linear map
$\mathcal{N}(X)$	null space of matrix X, defined as $\{u \mid Xu = 0\}$
$\mathcal{A} \subseteq \mathcal{B}$	$\mathcal{A}$ is a subset of $\mathcal{B}$
cenv f	convex envelope of function$f$
k	desired rank of a solution, $k \in \mathbb{N}_+$
G	Gram matrix defined as $G = X^T X$
P	permutation matrix
$\mathcal{P}$	feasibility set of (1)
A, B, C, W	given data of (1)
n, m, p, q	numbers of rows/columns of the data
$f(X)$	objective function of (1)
$g(X)$	linear objective function of a general rank-constrained problem (30)
α	relative weight of bi-criterion problems
γ	parameter of modified methods
M	maximum number of allowed "constant-rank" iterations in algorithms
$\varepsilon > 0$	tolerance constant for the numerical rank
$\delta > 0$	small regularization constant for log-det heuristic
$\rho > 0$	tolerance constant for stopping criterion of the bisection algorithm

Introduction

Matrix approximation problems search for the best possible approximation of a given matrix by solving a minimization program, where the objective function uses a specific matrix norm and the constraints represent requirements put onto the matrix variable. Besides a large scale of standard matrix approximation problems, this thesis also covers matrix completion problems and well-known Procrustes problems. The significance of solving matrix approximation problems lies in their diverse applications in fields such as data analysis, engineering, machine learning, computer vision, signal processing, and finance. Therefore, analyzing these problems and refining their solution algorithms can contribute to the advancement of these relevant fields.

Norm minimization problems have been studied since the nineteenth century when Legendre [67] and Gauss [49] introduced the first methods to minimize least squares, which is now a standard method in regression analysis. Today, unconstrained norm minimization problems can be solved by many modern unconstrained optimization techniques. Taking into account specific norms, even closed-form solutions are known [19, §1.2]. Norm minimization problems that involve linear or semidefinite constraints can be solved using interior point methods, which were first introduced by Karmarkar [63] in 1984 for linear programming and have since been extensively studied. Efficient polynomial-time algorithms have also been introduced for semidefinite programs [4, 78, 64] and are now implemented in numerous solvers [8, 58, 56, 55]. It is common for the feasibility set of a norm minimization problem to involve even quadratic constraints, which can be either directly rewritten or relaxed as semidefinite constraints.

In this thesis, we also deal with low-rank matrix approximations, which are an active research area. The rank constraint often arises in real-life applications, since it can model the inherent structure or complexity of data or processes [29]. However, solving rank-constrained matrix approximation problems can be challenging as they are nonconvex and NP-hard ([36, 91]). Since the twentieth century, they have been solved using methods based on singular value decomposition [86, 32], later by alternating projections [22]. In the early 2000s, rank minimization heuristics and rank reduction algorithms were designed to work in practice [37, 27, 68]. In recent years, exact algorithms based on symbolic

computation [77] and mixed projections [13] have gained popularity, underscoring the significance of this topic in modern research.

In general, matrix approximation problems have not been analyzed in a unified framework due to the diverse structure and properties of particular subclasses. Existing algorithms designed for specific subclasses often cannot be extended to solve other subclasses, resulting in numerous subclasses that lack a solution algorithm. Therefore, the main contribution of this thesis is the introduction of a unified conic optimization framework for solving matrix approximation problems constrained by linear, semidefinite, quadratic, or rank constraints. Although the conic optimization approach may not be as effective as methods tailored to specific subclasses, it offers the advantage of also covering nontrivial subclasses, such as weighted, and oblique Procrustes problems. Furthermore, unlike the existing approaches, the proposed conic optimization framework is not limited to a specific choice of the matrix norm in the objective, highlighting its potential to contribute to advancements in the field of matrix approximation problems.

As a consequence, our objective is to demonstrate the performance of the proposed approach in solving selected subclasses that arise in applications, such as the problem of finding the nearest low-rank correlation matrix and various types of Procrustes problems. Additionally, as the proposed approach may require solving rank-constrained optimization problems, our secondary objective is to modify the existing algorithms and design a new solution algorithm. We evaluated the performance of the proposed algorithm through extensive numerical experiments.

This thesis is divided into five chapters. The first chapter introduces a generalized formulation of matrix approximation problems, which also covers matrix completion problems and Procrustes problems. In this chapter, we provide three practical examples that fit the structure of the matrix approximation problem and serve as motivation for our research.

The second chapter presents a brief summary of the theoretical background of conic linear programming and introduces transformations to deal with norm minimization and quadratic constraints of the matrix approximation problem in a conic optimization frame-

work. These transformations lead to the formulation of semidefinite problems, which may also include an additional rank constraint.

In the third chapter, we discuss several techniques to address the rank constraint in otherwise convex problems. In addition, we provide an overview of existing algorithms, discuss their drawbacks, and propose a bisection algorithm for solving convex problems with an additional rank constraint.

The last two chapters focus on applications. The fourth chapter applies the proposed conic optimization approach to low-rank matrix approximation problems, specifically the problem of finding the nearest low-rank correlation matrix. We demonstrate the performance of the newly designed bisection algorithm in solving the corresponding rank-constrained semidefinite problems. Some of the numerical results described in this chapter were presented at the Algoritmy 2020 conference held in Podbanské and published in [44].

The fifth chapter discusses the constrained Procrustes problems that commonly occur in numerous applications. We show how the introduced conic approach allows solving various subclasses of Procrustes problems exclusively using conic optimization tools, including some nontrivial subclasses. The results summarized in this chapter were presented at the MMEI 2021 (Conference on Mathematic Methods in Economy and Industry) in Smolenice [45] and at the ODS 2022 (Conference on Optimization and Decision Science) in Florence [46]. These results have been submitted as an arXiv preprint [47].

1 Motivation

Matrix approximation problems are an attractive field of study with a wide range of practical applications in various disciplines, such as data science, engineering, signal processing, machine learning, or finance. In these fields, data is often represented as matrices that may not possess desired properties or may have missing entries. As a result, it becomes necessary to approximate a given matrix with another matrix that satisfies specific constraints or estimate the missing entries to achieve a resulting matrix that is as close as possible to the original matrix in terms of a matrix norm.

In this thesis, we study a generalized formulation of matrix approximation problems of the form

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & f(X) := \|W \circ (C - AXB)\| \\ & X \in \mathcal{P}, \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{p \times m}, B \in \mathbb{R}^{n \times q}, W, C \in \mathbb{R}^{p \times q}$ are the data and $X \in \mathbb{R}^{m \times n}$ is the matrix variable. Here $\circ$ denotes the Hadamard (element-wise) product (see Appendix A.2), and the matrix W specifies the target, that is,

$$W_{ij} = \begin{cases} 1, & \text{if } C_{ij} \text{ is given,} \\ 0, & \text{if } C_{ij} \text{ is missing.} \end{cases} \tag{2}$$

Clearly, if all elements of W are equal to 1, then the objective of (1) can be written as $\|C - AXB\|$.

In general, the formulation of the problem (1) corresponds to a matrix approximation problem, where the matrix variable X is assumed to belong to a feasible set $\mathcal{P}$. To specify the feasible set $\mathcal{P}$, we will assume the following throughout the thesis:

Assumption 1.1. *The feasible set $\mathcal{P}$ in* (1) *is described by linear, semidefinite, quadratic, or rank constraints.*

Even though the assumption might seem restrictive, it covers the well-known classes of Procrustes problems, such as orthogonal and oblique Procrustes problems, but allows for many different classes of matrix approximation problems widely studied in the literature.

In many research papers, the Frobenius norm is commonly used in the objective of (1) to compare matrices and measure their similarity. However, some authors also consider

other matrix norms, such as the l_1 norm and the l_∞ norm, which are less sensitive to outliers and are considered more robust alternatives to least squares. Additionally, our generalized formulation (1) also encompasses the spectral norm, which provides a way to quantify the similarity of the matrices in terms of their eigenvalues. Despite its usefulness in applications, the spectral norm can be challenging to handle using standard approaches.

In the following examples, we present formulations of several practical problems that may serve as a motivation for our research.

Example 1.1. ***Approximating correlation matrix in finance.*** Correlation matrices are frequently used in finance for tasks such as portfolio optimization, risk management, and data analysis. However, in these fields, data are rarely available for all observed time points, such as prices of stocks traded on different exchanges. For example, consider a data set presented in Table 1, which includes the prices of eight stocks observed at ten points in time, where NaNs represent missing values.

In this scenario, an empirical correlation matrix is computed for each element separately to minimize the number of neglected observations. As a result, we obtain the following empirical correlation matrix

$$C = \begin{pmatrix} 1 & -0.323 & 0.146 & 0.553 & -0.252 & 0.201 & -0.033 & -0.241 \\ -0.323 & 1 & 0.260 & 0.140 & 0.573 & 0.015 & 0.269 & 0.282 \\ 0.146 & 0.260 & 1 & -0.060 & 0.788 & 0.774 & -0.718 & 0.910 \\ 0.553 & 0.140 & -0.060 & 1 & -0.006 & 0.074 & 0.499 & -0.230 \\ -0.252 & 0.573 & 0.788 & -0.006 & 1 & 0.890 & -0.220 & 0.881 \\ 0.201 & 0.015 & 0.774 & 0.074 & 0.890 & 1 & -0.193 & 0.822 \\ -0.034 & 0.269 & -0.718 & 0.499 & -0.220 & -0.193 & 1 & -0.537 \\ -0.241 & 0.282 & 0.910 & -0.230 & 0.881 & 0.822 & -0.537 & 1 \end{pmatrix}.$$

Before continuing, we present the definition of the correlation matrix.

Definition 1.1. *A correlation matrix is a positive semidefinite matrix with a unit diagonal, whose element (i, j) denotes the correlation between variables x_i and x_j.*

After inspecting the eigenvalues of C (-0.13, -0.04, 0.05, 0.23, 0.63, 1.63, 1.74, 3.89), it becomes apparent that the empirical correlation matrix is not semidefinite. As a result, it does not meet the criteria for a correlation matrix, as defined in Definition 1.1. Instead, it only serves as an approximation of the actual correlation matrix. Therefore, the problem of finding the nearest correlation matrix must be addressed, as suggested in [60, 7]. The task is to find a correlation matrix $X \in \mathbb{S}^n$ that satisfies Definition 1.1 and serves as the

59.875	42.734	47.938	60.359	NaN	69.625	61.500	62.125
53.188	49.000	39.500	64.813	34.750	56.625	83.000	44.500
55.750	50.000	38.938	62.875	30.188	43.375	NaN	29.938
65.500	51.063	45.563	69.313	48.250	62.375	85.250	46.875
69.938	47.000	52.313	71.016	37.500	59.359	61.188	48.219
61.500	44.188	NaN	57.000	35.313	55.813	51.500	62.188
59.230	48.210	62.190	61.390	54.310	70.170	61.750	91.080
NaN	48.700	60.300	68.580	61.250	70.340	61.590	90.350
52.900	52.690	54.230	61.670	68.170	NaN	57.870	88.640
57.370	59.040	59.870	62.090	61.620	66.470	65.370	85.840

Table 1: Stock prices with missing values[1].

best approximation of the empirical correlation matrix C in terms of the Frobenius norm. This problem can be formulated as follows

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & \|C - X\|_F \\ & X_{ii} = 1, \quad i = 1, ..., n, \\ & X \succeq 0. \end{aligned} \tag{3}$$

This concept can also be extended to address the problem of completing the empirical correlation matrix C by incorporating the matrix $W \in \mathbb{R}^{n \times n}$, as defined in (2), into the objective of (3). This means that the objective would be in the form of $\|W \circ (C - X)\|_F$, where W ensures that the known entries of C do not change significantly in the approximation X. The Frobenius norm is the most suitable for comparing specific elements of the matrix $C - X$ that represent differences in specific correlations.

A more interesting problem arises in financial factor models, where the rank of the correlation matrix should not exceed the number of factors. For example, if we handle a model with k factors, the rank should be at most k (see [57, 106]). Given a real symmetric matrix C of order n, the problem of finding the nearest correlation matrix with the desired rank $k \in \mathbb{N}_+$ can be formulated as a rank-constrained optimization problem of the form

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & \|C - X\|_F \\ & X_{ii} = 1, \quad i = 1, ..., n, \\ & X \succeq 0, \\ & rank(X) \leq k. \end{aligned} \tag{4}$$

[1]Data available at `https://www.mathworks.com/help/stats/nearcorr.html`.

While (3) is a convex problem that can be easily solved by available solvers, the additional rank constraint in (4) leads to a nonconvex formulation, which requires special solution algorithms. We analyze the problem of finding the nearest low-rank correlation matrix in more detail in Chapter 4.

Example 1.2. ***Orthogonal transformation of an ancient map.*** Following [25, 53], the objective of this analysis is to assess the precision of John Speed's ancient map (see Figure 1) compared to a modern map from the Landranger series of Ordnance Survey Maps. Both maps show the locations of 20 towns and villages in the Worcestershire region. The locations were measured relative to the lower left corner of the maps, as provided in the data source[2]. The locations are visually compared in Figure 2.

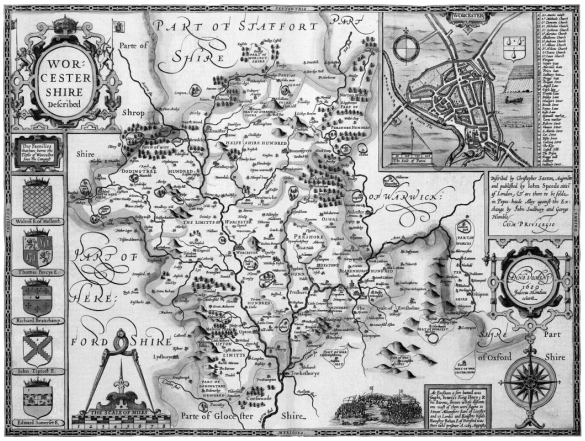


Figure 1: John Speed's historical map of the Worcestershire region in England. It was engraved and printed in 1611-1612[3].

Based on the observations from Figure 2, it is evident that the data points are shifted and slightly rotated. Aware of this, we can formulate the corresponding Procrustes problem as follows

$$\begin{aligned} \min_{X,d,\rho} \quad & f(X,d,\rho) := \|C - (\mathbf{1}_2 d^T + \rho A X)\|_F \\ & XX^T = I_2, \end{aligned} \tag{5}$$

where $X \in \mathbb{R}^{2\times 2}$ is a matrix variable representing an orthogonal transformation of the data, $d \in \mathbb{R}^2$ is a vector variable representing data translation, $\rho \in \mathbb{R}$ is a variable representing the scaling factor, $C \in \mathbb{R}^{20\times 2}$ denotes locations from the modern map and $A \in \mathbb{R}^{20\times 2}$ locations from the ancient map.

[2]Data available at `https://www.stata.com/manuals14/mvprocrustes.pdf`.

[3]Picture downloaded from `http://www.oldtowns.co.uk/`.

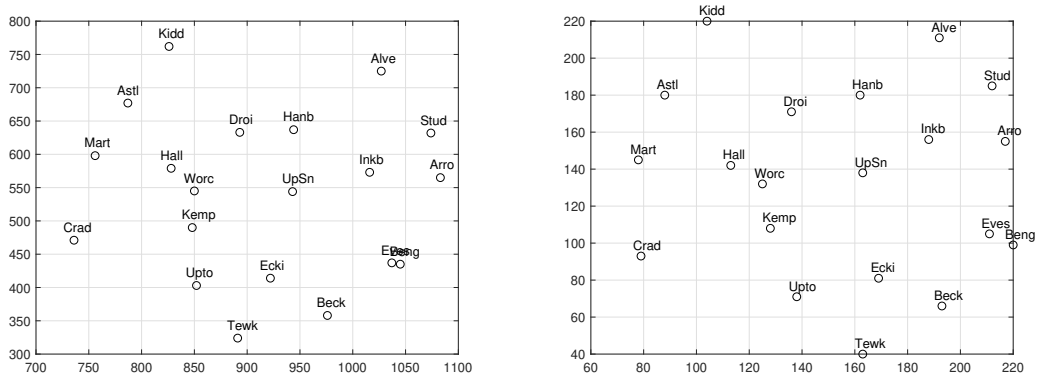


Figure 2: Locations. On the left: Locations from the ancient map. On the right: Locations from the modern map.

As shown in [53], the problem (5) can be solved in 3 steps:

1. find the orthogonal transformation X^* by solving an orthogonal Procrustes problem of the form

$$\begin{aligned} \min_{X \in \mathbb{R}^{2\times 2}} \quad & \|C - AX\|_F \\ & XX^T = I_2, \end{aligned} \tag{6}$$

2. find the scaling factor $\rho^*(X^*) = \underset{\rho\in\mathbb{R}}{\operatorname{argmin}}\, f(X^*, d^*(X^*, \rho), \rho)$,

3. find the translation vector $d^*(X^*, \rho^*) = \underset{d\in\mathbb{R}^2}{\operatorname{argmin}}\, f(X^*, d, \rho)$.

Optimality conditions can be used to derive explicit formulas for the optimal values of ρ^* and d^*. For fixed X and ρ, one obtains the following:

$$\rho^* = \frac{tr[JC(AX)^T]}{tr[JAX(AX)^T]}, \tag{7}$$

$$d^* = \frac{1}{p}(C - \rho AX)^T \mathbf{1}, \tag{8}$$

where $J = I_{20} - \frac{1}{20}\mathbf{1}\mathbf{1}^T$. As a result, the value of the objective function can be used to quantify the error of the ancient map.

As we discuss in later sections, the orthogonal Procrustes problem of the form (6) is easily solvable. However, slight modifications to the assumptions for the matrix variable can lead to more challenging problems, which we analyze in Chapter 5.

Example 1.3. ***Orthogonal least squares regression for feature extraction.*** Orthogonal least squares regression is a regression technique used in linear discriminant

Figure 3: Examples of images from the Yale data set[4].

analysis that involves finding an orthogonal transformation matrix $X \in \mathbb{R}^{m\times n}$ to project high-dimensional data (with dimension m) into a lower-dimensional space (with dimension $n << m$). It is used in machine learning and data analysis to identify a subset of features that are most informative for predicting the target variable while ensuring that the selected features are orthogonal to each other. According to [107], an orthogonal transformation matrix can preserve more information about the local structure, making the orthogonal least squares regression a useful tool in various applications such as feature selection, dimensionality reduction, and pattern recognition.

Orthogonal least squares regression is formulated as an orthogonal Procrustes problem of the form ([104])

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}} \quad & \|C - AX\|_F \\ & X^T X = I_n, \end{aligned} \tag{9}$$

where $C \in \mathbb{R}^{p\times n}$ and $A \in \mathbb{R}^{p\times m}$ represent the given data.

Consider the Yale data set, consisting of 165 gray-scale images of 15 individuals, with each individual having 11 images, representing different facial expressions or configurations. Figure 3 shows several samples of this data set. The task is to identify the most important facial features that predict the identity of the individual, such as the positions of certain landmarks on the face or the intensities of certain regions. To achieve this, the orthogonal least squares regression can be used to extract the most informative features that are correlated with the identity labels of individuals.

To perform feature extraction, we follow the approach described in [104]. Consider a data set $S = [s_1, ..., s_p] \in \mathbb{R}^{m\times p}$, which contains p samples with m features drawn

[4]Data sourced from `https://www.kaggle.com/datasets/olgabelitskaya/yale-face-database`.

from n classes. In the Yale data set, we have $p = 165$ images (samples) with $m = 256$ features corresponding to $n = 15$ individuals. Let $K = [k_1, ..., k_p] \in \mathbb{R}^{n\times p}$ be the class indicator matrix. This means that if the image s_i belongs to the j-th individual, then $k_i = e_j$, where $e_j \in \mathbb{R}^n$ is the j-th column of the standard basis. The model includes an orthogonal transformation matrix $X \in \mathbb{R}^{m\times n}$ and an associated bias $b \in \mathbb{R}^n$. Both X and b are determined using orthogonal least squares regression, which is formulated as an orthogonal Procrustes problem of the form ([104])

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^n} \quad & h(X,b) \coloneqq \|S^T X + \mathbf{1}_p b^T - K^T\|_F \\ & X^T X = I_n, \end{aligned} \tag{10}$$

where $S \in \mathbb{R}^{m\times p}$ and $K \in \mathbb{R}^{n\times p}$ are the given data described above.

Using the partial derivative of $h(X, b)$ with respect to b, we can express b as $b = \frac{1}{p}(K\mathbf{1}_p - X^T S\mathbf{1}_p)$. Consequently, the formulation (10) is simplified to the orthogonal Procrustes problem of the form (9), where $A = (I_p - \frac{1}{p}\mathbf{1}_p\mathbf{1}_p^T)S^T$ and $C = (I_p - \frac{1}{p}\mathbf{1}_p\mathbf{1}_p^T)K^T$.

Once the orthogonal Procrustes problem (9) has been solved and the most informative features have been selected, they can be used in various machine learning applications, such as classification, clustering, or dimensionality reduction. It is important to note that the orthogonal least squares regression (9) handles a rectangular orthogonal matrix variable $X \in \mathbb{R}^{m\times n}$, which makes it more challenging to solve compared to the orthogonal Procrustes problem (6) from Example 1.2, as we analyze in Chapter 5.

2 Conic optimization tools for norm minimization and quadratically constrained problems

In this chapter, we provide an overview of fundamental aspects of conic optimization, drawing from publications such as [19, 66, 8, 11, 100]. We then focus on norm minimization and quadratically constrained problems and propose transformations to suit a conic structure. Subsequently, we apply these results to solve a generalized matrix approximation problem (1) exclusively using conic optimization tools. For the sake of simplicity, we assume that $\mathcal{P}$ in (1) does not include rank constraints, which are analyzed separately in Chapter 3.

2.1 Convex optimization and conic linear programming

Convex optimization is a fundamental field of applied mathematics that deals with optimizing convex functions over convex sets. Convexity is a crucial property for developing computationally efficient algorithms such as interior point methods (see [4, 78, 64]), making it valuable to formulate real-life problems as convex. As a result, convex optimization has become an essential tool for researchers, practitioners, and decision-makers across a wide range of fields, such as control theory, combinatorial optimization, engineering, computer science, statistics, finance, and more.

Conic linear programming is a general framework for modeling all convex optimization problems. Besides standard convex problems, it covers even specific classes of convex problems that cannot be formulated in a traditional way (e.g. the class of semidefinite or copositive programs). The conic structure has significantly influenced the development of interior point methods, which are now implemented in various optimization software packages, such as Mosek [8], Gurobi [58], Sedumi [89], SDPT3 [95], CVXOPT [5], CVXR [43], CPLEX [26] and others. In our research, we used the SDPT3 solver implemented in CVX, a package for specifying and solving convex programs [56, 55]. It is worth mentioning that the efficiency of a method is typically measured by the running time of the code with the implemented method, which reflects the number of elementary operations performed during the optimization process.

In the realm of convex optimization, convex cones play an essential role. In this context, we will define convex cones in the vector space $\mathbb{R}^n$. Additionally, we present the standard formulation of a linear conic programming problem and discuss four fundamental types of convex cones.

Definition 2.1 ([19, §2.1.5]). *A set $\mathcal{K}$ is called a convex cone if for any $x_1, x_2 \in \mathcal{K} \subseteq \mathbb{R}^n$ and $\alpha, \beta \geq 0$ we have $\alpha x_1 + \beta x_2 \in \mathcal{K}$.*

Given a closed convex cone $\mathcal{K}$, the conic linear programming problem is standardly ([19, §2.1.5],[8, §1]) formulated as follows

$$\begin{array}{lrcl} \underset{x \in \mathbb{R}^n}{min} & c^T x & & \\ & Ax & = & b, \\ & x & \in & \mathcal{K}, \end{array} \tag{11}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

Note that the nonnegative orthant $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n \mid x \geq 0\}$ is also a convex cone. Therefore, linear programming (LP), as the historically oldest and structurally simplest class of optimization problems, is a specific subclass of conic optimization. Since conic problems maintain the linear objective and linear constraints, they are similar to LP problems up to the fact that the variable can belong to a convex cone other than the nonnegative orthant $\mathbb{R}^n_+$. In this sense, a conic problem can be perceived as a generalization of linear optimization. However, unlike LP problems, a general conic linear program (11) can also handle nonlinear constraints through the associated convex cone $\mathcal{K}$. This allows it to capture a wide range of applications that cannot be treated with LP, making it a versatile tool for optimization in various fields.

According to [8], there are four basic types of convex cones that allow the formulation of various types of nonlinear constraints and suffice to express most of the convex optimization problems encountered in practical applications:

- quadratic cone

$$\mathcal{Q}^n = \{x \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + x_3^2 + ... + x_n^2}\},$$

- power cone

$$\mathcal{P}_n^{\alpha, 1-\alpha} = \{x \in \mathbb{R}^n \mid x_1^\alpha x_2^{1-\alpha} \geq \sqrt{x_3^2 + ... + x_n^2}, x_1, x_2 \geq 0\}, \text{ where } 0 < \alpha < 1,$$

- exponential cone

$$\mathcal{K}_{exp} = \{(x_1, x_2, x_3) \mid x_1 \geq x_2 e^{\frac{x_3}{x_2}}, x_2 > 0\} \cup \{(x_1, 0, x_3) \mid x_1 \geq 0, x_3 \leq 0\},$$

- semidefinite cone

$$\mathbb{S}^n_+ = \{X \in \mathbb{S}^n \mid z^T X z \geq 0, \forall z \in \mathbb{R}^n\}.$$

The class of quadratic cone programming, known as second-order cone programming (SOCP), covers a large scale of problems involving absolute values, the Euclidean norm, convex quadratic sets, ellipsoidal sets, and more (see [3], [70], [8, §3]). Power cones are a generalization of quadratic cones and provide a convenient way to handle constraints involving powers other than two. They can be used to express various types of constraints, such as p norms and geometric mean (see [8, §4]). The exponential cone is useful for dealing with constraints involving exponentials and logarithms, which commonly arise in portfolio optimization, entropy problems, logistic regression, or geometric programming (see [8, §5]). In the thesis, we pay special attention to the semidefinite cone $\mathbb{S}^n_+$, which we characterize in a dedicated subsection below.

2.1.1 Semidefinite programming

Semidefinite programming (SDP) deals with conic linear programs (11), where $\mathcal{K}$ is the cone of positive semidefinite matrices $\mathbb{S}^n_+$ in the subspace of symmetric matrices $\mathbb{S}^n$. This means that SDP problems minimize a linear objective over the intersection of the linear space and $\mathbb{S}^n_+$. A standard SDP problem is formulated in the form ([19, §4.6], [27, §4.1.1])

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & tr(CX) \\ & tr(A_i X) = b_i,\ i = 1, ..., m, \\ & X \succeq 0, \end{aligned} \tag{12}$$

where $C, A_1, ..., A_m \in \mathbb{S}^n$ and $b \in \mathbb{R}^m$. Note that $tr(CX)$ represents the vector inner product for matrices, that is

$$tr(CX) = \langle C, X \rangle = svec(C)^T svec(X), \tag{13}$$

where $svec(Y)$ denotes symmetric vectorization of $Y \in \mathbb{S}^n$ defined as

$$svec(Y) = \begin{pmatrix} Y_{11} & \sqrt{2}Y_{12} & Y_{22} & \sqrt{2}Y_{13} & \sqrt{2}Y_{23} & Y_{33} & \dots & Y_{nn} \end{pmatrix}^T \in \mathbb{R}^{\frac{n(n+1)}{2}}. \tag{14}$$

Another common form of the SDP problem is its dual formulation that covers constraints formulated as linear matrix inequalities (LMIs). It is formulated as follows

$$\begin{aligned} \underset{y\in\mathbb{R}^m}{max} \quad & b^T y \\ \sum_{i=1}^{m} y_i A_i \quad & \preceq \quad C. \end{aligned} \tag{15}$$

SDP remains relevant due to its ability to model or approximate a wide range of practical problems, including control theory, combinatorial optimization, spectral analysis, statistical experimental design, machine learning, and finance. Furthermore, SDP allows for the effective handling of eigenvalues, which is of significant importance (see Appendix B.2).

There are various classes of optimization problems that can be represented as SDP programs through specific transformations. It is known that convex quadratic constraints can be directly reformulated as semidefinite constraints, while nonconvex quadratic constraints can be relaxed by semidefinite constraints. In the thesis, we encounter the need to handle the norm in the objective of the generalized matrix approximation problem (1), as well as different types of constraints allowed to define the feasibility set $\mathcal{P}$ of (1). To address this, we present the relevant transformations in the following subsections.

2.2 Quadratically constrained problems

In this section, our aim is to introduce a conic optimization approach to solve the generalized matrix approximation problem (1). First, we want to deal with all types of constraints allowed to define the feasibility set $\mathcal{P}$.

For the sake of simplicity, let us neglect the objective and consider the feasibility problem of the form

$$\begin{aligned} \text{find} \quad & X \in \mathbb{R}^{m\times n} \\ & X \in \mathcal{P}. \end{aligned} \tag{16}$$

If the feasible set $\mathcal{P}$ consists only of linear or semidefinite constraints, problem (16) is an LP or SDP problem, which can be efficiently solved, as discussed above. In addition to linear and semidefinite constraints, we also assume quadratic constraints and rank constraints that define $\mathcal{P}$.

In this section, we investigate the quadratic constraints. As known from the convex analysis [19], the convex quadratic constraints can be represented as semidefinite constraints after using the Schur complement lemma. We provide its statement below.

Lemma 2.1 ([103, §6.3]). *Let*

$$M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

be a symmetric matrix with the $k \times k$ block A and the $l \times l$ block C.

If A is positive definite, then M is positive (semi)definite if and only if the matrix $C - B^T A^{-1} B$ (called the Schur complement of A in M) is positive (semi)definite.

If C is positive definite, then M is positive (semi)definite if and only if the matrix $A - BC^{-1}B^T$ (called the Schur complement of C in M) is positive (semi)definite.

Now, if we take, for example, a convex quadratic constraint of the form $X^T X \preceq G$, using Lemma 2.1, it can be rewritten as

$$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0, \tag{17}$$

because the matrix $G - X^T X$ is the Schur complement of the identity matrix in V.

A more complicated situation occurs when dealing with nonconvex quadratic constraints. In such cases, the rank of a matrix variable must also be taken into account. The following lemma provides a brief reminder of how the rank of the block matrix (17) is determined.

Lemma 2.2 ([27, §A.4]). *Let A and C be symmetric matrices, while A is invertible. Then the rank of the block matrix E can be determined as follows*

$$rank\left(\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}\right) = rank\left(\begin{bmatrix} A & 0 \\ 0^T & C - B^T A^{-1} B \end{bmatrix}\right) = rank\,(A) + rank\left(C - B^T A^{-1} B\right). \tag{18}$$

The following proposition states an important property of positive semidefinite matrices. A similar statement can be found in [91]. We will provide proof of this proposition based on Lemma 2.1 and Lemma 2.2.

Proposition 2.1. *Let $G \in \mathbb{S}^n$ and $X \in \mathbb{R}^{m \times n}$. Then*

$$G = X^T X \quad \Leftrightarrow \quad \left[G \succeq X^T X \ \wedge \ rank\left(\begin{bmatrix} I_m & X \\ X^T & G \end{bmatrix}\right) = m \right]. \tag{19}$$

Proof. ($\Rightarrow$): If $G = X^T X$, then $G \succeq X^T X$ and $rank(G - X^T X) = 0$. Therefore, due to Lemma 2.2 we have

$$rank\left(\begin{bmatrix} I_m & X \\ X^T & G \end{bmatrix}\right) = m.$$

($\Leftarrow$): If we assume that $G - X^T X \succeq 0$, we can use Lemma 2.1 to equivalently rewrite this matrix inequality as

$$Z = \begin{bmatrix} I_m & X \\ X^T & G \end{bmatrix} \succeq 0. \tag{20}$$

The rank of the block matrix Z is calculated according to Lemma 2.2 as follows

$$\begin{aligned} rank\left(\begin{bmatrix} I_m & X \\ X^T & G \end{bmatrix}\right) = rank\left(\begin{bmatrix} I_m & 0 \\ 0^T & G - X^T X \end{bmatrix}\right) &= rank\left(I_m\right) + rank\left(G - X^T X\right) \\ &= m + rank\left(G - X^T X\right). \end{aligned}$$

As the rank of the block matrix Z is m, the rank of the matrix $G - X^T X$ must be 0, implying that $G - X^T X = 0$. Thus, $G = X^T X$. □

The statement of Proposition 2.1 offers a representation of a nonconvex quadratic constraint in the form $G = X^T X$. Moreover, this representation can be extended to derive representations of other types of nonconvex quadratic constraints. Table 2 provides a summary of various types of quadratic constraints, along with their representations using linear, semidefinite, and rank constraints. These representations are derived from established matrix analysis results [19, 27], and can be obtained using statements of Lemma 2.1 and Proposition 2.1, as we present in Appendix B.4.

2.3 Norm minimization problems

In this section, we handle a more general formulation of norm minimization problems of the form

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & f(X) := \|\mathcal{L}(X)\| \\ \text{s.t.} \quad & X \in \mathcal{P}, \end{aligned} \tag{21}$$

where $\mathcal{L} : \mathbb{R}^{m \times n} \to \mathbb{R}^{p \times q}$ is a linear map and $\mathcal{P}$ is the feasible set satisfying Assumption 1.1. The generalized matrix approximation problem (1) fits this structure with $\mathcal{L}(X) \coloneqq W \circ (C - AXB)$.

Constraint	Representation		Variables
$X^T X \preceq G$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$		X, V
$X^T X = G$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$	$rank(V) = m$	X, V
$X^T X \succeq G$	$V = \begin{pmatrix} I_m & X \\ X^T & Y \end{pmatrix} \succeq 0$	$rank(V) = m$, $Y - G \succeq 0$	X, V, Y
$tr(X^T X) \leq g$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$	$tr(G) \leq g$	X, V, G
$tr(X^T X) = g$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$	$tr(G) = g$, $rank(V) = m$	X, V, G
$tr(X^T X) \geq g$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$	$tr(G) \geq g$, $rank(V) = m$	X, V, G
$diag(X^T X) \leq h$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$	$diag(G) \leq h$, $rank(V) = m$	X, V, G
$diag(X^T X) = h$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$	$diag(G) = h$, $rank(V) = m$	X, V, G
$diag(X^T X) \geq h$	$V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0$	$diag(G) \geq h$, $rank(V) = m$	X, V, G

Table 2: Quadratic constraints representation via semidefinite, linear, and rank constraints. Quadratic constraints are given by $m \times n$ matrix X, $n \times n$ matrix G, n-dimensional vector h and scalar g.

matrix norm	notation	definition
l_1 norm	$\|Y\|_1$	$\max\limits_{1\leq j\leq n} \sum_{i=1}^{m} \|Y_{ij}\|$
l_∞ norm	$\|Y\|_\infty$	$\max\limits_{1\leq i\leq m} \sum_{j=1}^{n} \|Y_{ij}\|$
l_2 /spectral norm	$\|Y\|_2$	$\sigma_{max}(Y) = \sqrt{\lambda_{max}(Y^TY)}$
Frobenius norm	$\|Y\|_F$	$\sqrt{tr(YY^T)} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} Y_{ij}^2}$

Table 3: Matrix norms definitions. Matrix norm definitions of $Y \in \mathbb{R}^{m\times n}$, where σ_{max} and λ_{max} denote the largest singular value and the largest eigenvalue, respectively.

Our aim is to reformulate the objective of (21) to align with a conic structure. In the objective, we consider these four matrix norms: l_1 norm, l_2 norm, l_∞ norm, and Frobenius norm. For each of these matrix norms, we derive a conic reformulation of the problem (21). To handle the l_1, l_2, and l_∞ norms, we employ standard transformation techniques as described in [19]. Additionally, we propose a novel approach for handling the Frobenius norm. The definitions of these matrix norms are summarized in Table 3.

As mentioned in Chapter 1, different matrix norms serve specific purposes and find applications in various scenarios. For instance, the l_1 norm is used to measure the "sparsity" of a matrix, penalizing large entries in any column. On the other hand, the l_∞ norm is useful for measuring the maximum absolute error. The l_2 norm, also known as a spectral norm, is frequently used to measure the "scale" of a matrix, as it is closely related to the eigenvalues of Y^TY. Meanwhile, the Frobenius norm is the most commonly used norm in real-world matrix approximation problems, as it measures the element-wise distance between two matrices.

In the following propositions, we summarize equivalent reformulations of the general norm minimization problem (21) with respect to particular norms. Detailed transformations for these problems can be found in Appendix B.3. It is worth noting that two optimization problems are considered equivalent if an optimal solution of one problem can be used to construct an optimal solution of the other problem.

Proposition 2.2. *The problem*

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & f(X) := \|\mathcal{L}(X)\|_1 \\ & X \in \mathcal{P} \end{aligned} \tag{22}$$

is equivalent to the problem

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}, t \in \mathbb{R}, S \in \mathbb{R}^{p \times q}} \quad & t \\ & X \in \mathcal{P}, \\ -S \leq & \mathcal{L}(X) \leq S, \\ & S^T \mathbf{1}_p \leq t \mathbf{1}_q. \end{aligned} \tag{23}$$

Proof. If $\hat{X}$ is optimal for (22), and we define $\hat{S} \in \mathbb{R}^{p \times q}$ such that $\hat{S}_{ij} := |\mathcal{L}(\hat{X})_{ij}|$ and $\hat{t} = \max_j \sum_{i=1}^{p} \hat{S}_{ij}$, then $(\hat{X}, \hat{S}, \hat{t})$ is feasible for (23) and $\hat{t} = f(\hat{X})$. Reversely, if (X^*, S^*, t^*) is optimal for (23), then X^* is feasible for (22). From the last constraint, we have $t^* \geq \max_j \sum_{i=1}^{p} S^*_{ij}$, and from the second constraint we have $S^*_{ij} \geq |\mathcal{L}(X^*)_{ij}|$. Consequently, using the l_1 norm definition from Table 3, we have the following

$$t^* \geq \max_j \sum_{i=1}^{p} S^*_{ij} = \|S^*\|_1 \geq \|\mathcal{L}(X^*)\|_1 = f(X^*).$$

To sum up, we have

$$f(X^*) \leq t^* \leq \hat{t} = f(\hat{X}) \leq f(X^*),$$

where the first inequality follows from the feasibility of t^* in (23), the second inequality follows from the optimality of t^* for (23), and the last inequality follows from the optimality of $\hat{X}$ for (22). Therefore, $f(X^*) = f(\hat{X}) = \hat{t} = t^*$. □

In summary, if the l_1 norm is present in the objective of the generalized matrix approximation problem (1), it can be replaced with a linear objective and linear constraints, resulting in an equivalent LP problem, assuming that only linear functions define $\mathcal{P}$.

Proposition 2.3. *The problem*

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & f(X) := \|\mathcal{L}(X)\|_\infty \\ & X \in \mathcal{P} \end{aligned} \tag{24}$$

is equivalent to the following problem

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n},t\in\mathbb{R},S\in\mathbb{R}^{p\times q}} \quad & t \\ & X \in \mathcal{P}, \\ -S \leq \mathcal{L}(X) & \leq S, \\ & S\mathbf{1}_q \leq t\mathbf{1}_p. \end{aligned} \tag{25}$$

Proof. The proof is analogous to the proof of Proposition 2.2. □

Similarly to the l_1 norm, if the l_∞ norm is present in the objective of the generalized matrix approximation problem (1), it can be replaced with a linear objective and linear constraints. Assuming that the feasibility set $\mathcal{P}$ is polyhedral, the problem (24) becomes an LP problem. More interesting cases arise when dealing with the spectral norm or the Frobenius norm in the objective. In the following subsections, we show that in such cases, the generalized matrix approximation problem (1) is equivalent to a problem with a linear objective and an additional SDP constraint.

Proposition 2.4. *The problem*

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}} \quad & f(X) := \|\mathcal{L}(X)\|_2 \\ & X \in \mathcal{P}. \end{aligned} \tag{26}$$

is equivalent to the problem

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n},s\in\mathbb{R}} \quad & s \\ & X \in \mathcal{P} \\ & \begin{pmatrix} sI_p & \mathcal{L}(X) \\ \mathcal{L}(X)^T & sI_q \end{pmatrix} \succeq 0. \end{aligned} \tag{27}$$

.

Proof. Let $\hat{X}$ be optimal for (26) and define $\hat{s} := \|\mathcal{L}(\hat{X})\|_2$. From the definition of the spectral norm (see Table 3), we obtain

$$(\hat{s})^2 = \lambda_{max}\Big(\mathcal{L}(\hat{X})\mathcal{L}(\hat{X})^T\Big),$$

which implies

$$(\hat{s})^2 I_q - \mathcal{L}(\hat{X})^T\mathcal{L}(\hat{X}) \succeq 0$$

and due to the Schur complement property from Lemma 2.1, such $(\hat{X}, \hat{s})$ satisfies the last constraint of (27). Therefore, it follows that $(\hat{X}, \hat{s})$ is feasible for (27) and $\hat{s} = f(\hat{X})$. On the reverse side, if (X^*, s^*) is optimal for (27), then X^* is feasible for (26). From the last constraint of (27) and the definition of the spectral norm, we have

$$s^* \geq \sigma_{max}\Big(\mathcal{L}(X^*)\Big) = \|\mathcal{L}(X^*)\|_2 = f(X^*).$$

In conclusion, it holds

$$f(X^*) \leq s^* \leq \hat{s} = f(\hat{X}) \leq f(X^*),$$

where the first inequality follows from the last constraint in (27), the second inequality follows from the optimality of s^* for (27), and the last inequality follows from the optimality of $\hat{X}$ for (26). Therefore, $f(X^*) = f(\hat{X}) = \hat{s} = s^*$. □

Proposition 2.5. *The problem*

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & f(X) := \|\mathcal{L}(X)\|_F^2 \\ & X \in \mathcal{P}. \end{aligned} \tag{28}$$

is equivalent to the problem

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}, Z \in \mathbb{S}^p} \quad & tr(Z) \\ & X \in \mathcal{P} \\ & \begin{pmatrix} I_q & \mathcal{L}(X)^T \\ \mathcal{L}(X) & Z \end{pmatrix} \succeq 0. \end{aligned} \tag{29}$$

Proof. Let $\hat{X}$ be optimal for (28), and define $\hat{Z} := \mathcal{L}(\hat{X})\mathcal{L}(\hat{X})^T$. Using the Schur complement property from Lemma 2.1, the last constraint of (29) can be rewritten as $Z \succeq \mathcal{L}(X)\mathcal{L}(X)^T$. Therefore, $(\hat{X}, \hat{Z})$ is feasible for (29) and $tr(\hat{Z}) = f(\hat{X})$. Reversely, if (X^*, Z^*) is optimal for (29), then X^* is feasible for (28). Due to the property of semidefinite matrices (see Lemma A.2), from the last constraint of (29) we have

$$tr(Z^*) \geq tr\Big(\mathcal{L}(X^*)\mathcal{L}(X^*)^T\Big) = \|\mathcal{L}(X^*)\|_F^2 = f(X^*).$$

Consequently, it holds

$$f(X^*) \leq tr(Z^*) \leq tr(\hat{Z}) = f(\hat{X}) \leq f(X^*),$$

where the first inequality follows from the last constraint in (29), the second inequality follows from the optimality of Z^* for (29), and the last inequality follows from the optimality of $\hat{X}$ for (28). Therefore, we have $f(X^*) = f(\hat{X}) = tr(\hat{Z}) = tr(Z^*)$. □

2.4 Summary

In the previous subsections, we have outlined how to address various types of objectives and constraints in the generalized matrix approximation problem (1). In Section 2.3, we discussed how transformations of the objective can result in a linear objective with an additional linear or semidefinite constraint. Moreover, in Section 2.2, we presented transformations for quadratic constraints that allow us to express the feasibility set $\mathcal{P}$ in terms of linear, semidefinite, and rank constraints.

In summary, we are equipped to handle either SDP problems or rank-constrained SDP problems, regardless of the types of objectives and constraints in (1). In the case of SDP problems, we can find their solution effectively. On the other hand, rank-constrained SDP problems remain nonconvex. However, their SDP structure is beneficial for constructing several solution methods to handle the nonconvex rank constraint. This topic is further analyzed in Chapter 3.

3 Rank-constrained optimization problems

The requirement to impose a rank constraint on a matrix variable arises in various areas, such as control, statistics, finance, engineering, or combinatorial optimization. For instance, a rank constraint can limit the number of parameters in a statistical model to fit a random process or ensure that a shape embeds in a low-dimensional space. In addition, it may emerge as a result of a transformation, as demonstrated in Proposition 2.1 when handling nonconvex quadratic constraints.

In this chapter, we delve into the realm of rank-constrained optimization problems, as defined in [91], where the goal is to optimize a convex objective function subject to a set of convex constraints along with rank constraints imposed on the matrix variable. The problem can be formulated as follows

$$\begin{aligned} \underset{X \in \mathbb{S}^n_+}{min} \quad & g(X) \coloneqq tr(CX) \\ & X \in \mathcal{C}, \\ & rank(X) \leq k, \end{aligned} \tag{30}$$

where $C \in \mathbb{S}^n$, $\mathcal{C}$ is a convex set, and $k \in \mathbb{N}_+$ is a desired rank of the matrix variable $X \in \mathbb{S}^n_+$. It is worth noting that when $\mathcal{C}$ is an affine set, the problem (30) corresponds to the rank-constrained SDP problem. Although transformations of the matrix approximation problem (1) introduced in Chapter 2 may lead specifically to the rank-constrained SDP reformulation, this chapter focuses on addressing a more general rank-constrained optimization problem (30), in order to present known results that are relevant to rank-constrained optimization problems in general.

As apparent from the structure, the rank constraint is the only source of nonconvexity in "otherwise convex" problem (30). Although the rank-constrained optimization problem (30) is NP-hard ([36, 91]), several algorithms designed for finding a feasible solution of the rank-constrained optimization problem (30) have been adjusted to find also an approximation of an optimal solution of the rank-constrained optimization problem (30). We discuss these methods in more detail in Section 3.3.1 and Section 3.3.2. In addition, there are several local search algorithms, as mentioned in [91, 9]. Efforts have also been made to develop exact algorithms for solving the rank-constrained optimization problem (30). However, in this thesis, we focus on heuristics and rank reduction algorithms

due to limitations of exact algorithms, such as their applicability only in small dimensions (e.g., [77]) or their recent development (e.g., [13]). Nonetheless, the strong interest in developing a framework for modeling and solving rank-constrained optimization problems highlights the advantages of reformulating real-life problems as rank-constrained optimization problems.

The proposal of solution methods was conditioned by some beneficial properties of the rank as a function of the symmetric positive semidefinite matrix variable. It is known ([27, §2.9.2.9]) that the rank is a quasiconcave function on the set of symmetric positive semidefinite matrices, which follows from the fact that

$$rank(X+Y) \geq min\{rank(X), rank(Y)\}, \text{ for } X, Y \in \mathbb{S}^n_+. \tag{31}$$

Furthermore, the diagonalizability of symmetric matrices enables the investigation of the relationship between their rank and eigenvalues. The following lemma summarizes two essential results known from matrix theory [19, 27, 103].

Lemma 3.1. *Let $X \in \mathbb{S}^n_+$, $k \leq n$ and $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ be eigenvalues of X. Then it holds*

$$\lambda_1 + \lambda_2 + ... + \lambda_k = tr(X) \;\; \Leftrightarrow \;\; rank(X) \leq k, \tag{32}$$

and

$$\lambda_{k+1} + \lambda_{k+2} + ... + \lambda_n = 0 \;\; \Leftrightarrow \;\; rank(X) \leq k. \tag{33}$$

It is known that the rank of a symmetric positive semidefinite matrix equals the number of its nonzero eigenvalues. For purposes of numerical computation, we can define an ε-rank of a symmetric positive semidefinite matrix, which considers eigenvalues with magnitudes greater than ε as nonzero eigenvalues, where ε is a small positive value.

Definition 3.1. *Given $X \in \mathbb{S}^n_+$ with eigenvalues $\lambda_1 \geq ... \geq \lambda_n \geq 0$, and $k \in \mathbb{N}_+$, for $\varepsilon > 0$ we define the ε-rank of X as the number of its eigenvalues greater than ε. That is*

$$\varepsilon\text{-}rank(X) = k \;\; \Leftrightarrow \;\; \lambda_1 > \varepsilon, ..., \lambda_k > \varepsilon, \lambda_{k+1} \leq \varepsilon, ..., \lambda_n \leq \varepsilon.$$

In the following sections, we will discuss several methods for finding optimal or feasible solutions of the rank-constrained optimization problem (30).

3.1 Convex relaxation

With an increasing number of problems having a nonconvex formulation, there is a growing need to find effective ways to solve them. The simplest way to deal with a nonconvex constraint is using a convex relaxation. Many relaxation methods have the common feature that they approximate the original nonconvex problem by its (in some sense) closest convex counterpart. Although the solution of the relaxed problem may not be optimal for the original problem, it is often sufficient for practical purposes. Moreover, it provides a better lower bound for the optimal value of the original problem, compared to its dual ([19, §5.2]).

The convex relaxation of the rank-constrained optimization problem (30) consists in omitting the rank constraint and solving a convex program of the form

$$\begin{array}{ll} \underset{X \in \mathbb{S}^n_+}{min} & g(X) \coloneqq tr(CX) \\ & X \in \mathcal{C}, \end{array} \tag{34}$$

where $C \in \mathbb{S}^n$ and $\mathcal{C}$ is a convex set. In case $\mathcal{C}$ is affine, the problem (34) is called a semidefinite relaxation of the rank-constrained optimization problem (30).

It is obvious that if a rank-k solution is found, it is an optimal solution of the original rank-constrained optimization problem (30). However, as proved in [102], the contemporary interior-point methods for solving convex problems converge to the solution of the highest feasible rank. Therefore, it is not always guaranteed that a low-rank solution is found, and other approaches must be considered.

A significant property of the convex relaxation (34) is that it provides a lower bound on the objective of the rank-constrained optimization problem (30). This is due to the fact that the convex relaxation minimizes the same objective function g but over a larger set of feasible solutions, which supersedes the feasible set of the rank-constrained optimization problem (30). Let X^* be an optimal solution of the rank-constrained optimization problem (30) with the optimal value $g^* \coloneqq g(X^*)$ and let X_0 be an optimal solution of the convex relaxation (34) with the optimal value $g_0 \coloneqq g(X_0)$. Then it holds

$$g^* \geq g_0. \tag{35}$$

Convex relaxations, or semidefinite relaxations in particular, are commonly used in control and combinatorial optimization [18, 79], and integer programming, mainly associ-

ated with partitioning, assignment, and ordering [84, 85]. The most well-known semidefinite relaxation of a rank-constrained problem was introduced for the famous MAX-CUT problem [51].

3.2 Methods for solving rank-constrained feasibility problems

Consider the rank-constrained optimization problem (30) as a feasibility problem of the form

$$\begin{array}{rrcl} \text{find} & X \in \mathbb{S}^n_+ & & \\ & X & \in & \mathcal{C}, \\ & rank(X) & \leq & k, \end{array} \tag{36}$$

where $\mathcal{C}$ is a convex set and $k \in \mathbb{N}_+$ is the desired rank.

If X^* is an optimal solution of the rank-constrained optimization problem (30) with the optimal value $g^* \coloneqq g(X^*)$ and $\hat{X}$ is a solution of the rank-constrained feasibility problem (36), which is a feasible solution of the rank-constrained optimization problem (30) that provides value $\hat{g} \coloneqq g(\hat{X})$. Then $\hat{g}$ gives an upper bound on the optimal value g^*. It holds

$$g^* \leq \hat{g}. \tag{37}$$

One of the possible approaches to solve the rank-constrained feasibility problems (36) is to use the so-called rank minimization heuristics, which were introduced in [36, 37] and were originally designed to tackle general rank minimization problems of the form

$$\begin{array}{rrcl} \min\limits_{X \in \mathbb{S}^n_+} & rank(X) & & \\ & X & \in & \mathcal{C}. \end{array} \tag{38}$$

Although originally assuming a matrix variable $X \in \mathbb{S}^n_+$, the rank minimization problem (38) also encompasses problems with general $X \in \mathbb{R}^{m \times n}$. In [37], a theorem was proposed that establishes every rank minimization problem with a general matrix variable can be equivalently rewritten as a rank minimization problem with a symmetric positive semidefinite variable (see Appendix D.1).

In the following, we outline four methods for solving the rank minimization problem (38), which involves finding a feasible solution for the rank-constrained optimization problem (30). Firstly, we present two rank minimization heuristics that have been introduced and enhanced in [36, 37, 39, 38]. Later, we describe two rank reduction algorithms from [68, 27].

3.2.1 Trace heuristic

An essential tool for relaxing nonconvex problems is the concept of the convex envelope of a function (Definition C.1), which serves as a good convex surrogate of a nonconvex function ([37, §1]). The convex envelope of the rank function on the set

$$\mathcal{U}_1 = \{X \in \mathbb{S}^n_+ \mid 0 \preceq X \preceq I\}$$

is the trace function, as stated in Theorem C.1. We provide an alternative proof of this statement in Appendix C.2. Since the trace is the convex envelope of the rank function on the set $\mathcal{U}_1$, it is its minorizing convex function, that provides the tightest global lower bound on rank among all convex approximations. For further details on the convex envelope, we refer to Appendix C.1.

The trace heuristic for the rank minimization problem (38) involves replacing the rank function with the trace function in the objective, leading to a convex problem

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n_+} & tr(X) \\ & X \in \mathcal{C}. \end{array} \tag{39}$$

An alternative perspective on how the trace heuristic works is offered in [36]. The trace function can be viewed as the sum of its eigenvalues $\lambda_1, ..., \lambda_n$, denoted as $\lambda(X) = (\lambda_1, ..., \lambda_n)^T$. Since the trace of a positive semidefinite matrix is equal to the sum of its eigenvalues (as stated in Lemma 3.1), and since $\lambda_i = |\lambda_i|$, for $\forall i = 1, ..., n$, the trace can be expressed as the l_1 norm of the vector $\lambda(X)$ as follows

$$tr(X) = \sum_{i=1}^{n} \lambda_i(X) = \sum_{i=1}^{n} |\lambda_i(X)| = \|\lambda(X)\|_1. \tag{40}$$

Minimizing the trace function is then related to minimizing the l_1 norm of the vector $\lambda(X)$. As a result, the trace heuristic (39) can be interpreted as l_1 regularization, where an optimal solution contains many zero elements. Additionally, since the zero elements of $\lambda(X)$ correspond to zero eigenvalues of X, it is expected that the optimal solution of a problem with objective (40) is a low-rank matrix.

The trace heuristic (39) is a popular approach for solving the rank minimization problem (38) due to the linearity of the trace function in X, making it a convex program that can be efficiently solved. It is known to work reliably in applications when solving the

rank minimization problem (38). However, it may find a solution with a higher rank than the desired rank k, which cannot be considered a solution of the rank-constrained feasibility problem (36). Furthermore, when minimizing the rank of a matrix with constant diagonal elements or a constant sum of diagonal elements, the trace heuristic (39) is equivalent to the convex relaxation (34). This highlights the need for more advanced iterative algorithms to solve the rank-constrained feasibility problem (36) in such scenarios.

3.2.2 *Log*-det heuristic

In [38], the so-called log-det heuristic was proposed as a method to enhance the performance of the trace heuristic. The main concept behind this heuristic is to approximate the rank minimization problem (38) with the problem

$$\begin{aligned} \underset{X \in \mathbb{S}_+^n}{min} \quad & \log\det(X + \delta I_n) \\ X \in \mathcal{C}, \end{aligned} \tag{41}$$

where the symbol $\log\det$ denotes the logarithm of the determinant of the given matrix, $\mathcal{C}$ is a convex set, and $\delta > 0$ is a small regularization constant.

In Figure 4, we can observe how the rank function is approximated with the trace and the $\log\det$ function in a one-dimensional case. The plot displays the rank function, the trace function, and the $\log\det$ function for a scalar $x \in \mathbb{R}$ satisfying $tr(x) = |x|$ and

$$rank(x) = \begin{cases} 0, & \text{for } x = 0, \\ 1, & \text{otherwise.} \end{cases} \tag{42}$$

Although the function $\log\det(X + \delta I_n)$ is concave (as stated in Lemma D.1), it can still be considered as a suitable substitution for the rank function as it is smooth and locally minimizable using this iterative method

$$\begin{aligned} X_t = \quad \underset{X \in \mathbb{S}_+^n}{\operatorname{argmin}} \quad & tr((X_{t-1} + \delta I_n)^{-1} X) \\ X \quad \in \quad & \mathcal{C}, \end{aligned} \tag{43}$$

where the function $\log\det(X + \delta I_n)$ is approximated by its first-order approximation. More details can be found in Appendix D.3.

In [38], it has been shown that the sequence $\{\log\det(X_t + \delta I_n)\}$ converges to the local minimum of the function $\log\det(X + \delta I_n)$. Notably, the problem (43) can be interpreted as a weighted trace minimization, where the weights are given by $W_t = (X_t + \delta I_n)^{-1}$.

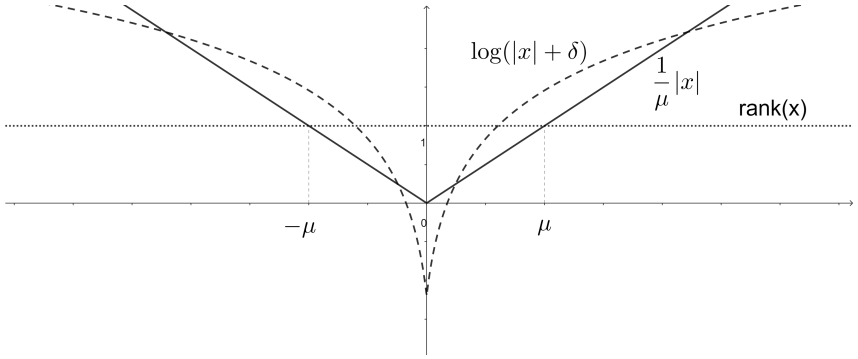


Figure 4: Illustration of the rank function approximation. The rank function is approximated by the trace function, and the function log det for a scalar $x \in \mathbb{R}$ according to [38].

At the beginning of the iterative process, we initialize $X_0 = I_n$, which ensures that the first iteration aligns with the trace heuristic (39). Then, the subsequent iterations aim to refine the solution obtained from the trace heuristic in order to achieve a lower-rank solution. From this perspective, the log-det heuristic (43) can be seen as an enhancement of the trace heuristic (39).

The iterative process (43) terminates when a rank-k solution is obtained. However, for numerical computation, we adjusted the stopping criterion to utilize the ε-rank (as defined in Definition 3.1), with a chosen value of ε. Additionally, in each iteration t, we determine the ε-rank of the current solution X_t and check if the number of iterations handling the constant-rank solutions does not exceed a chosen threshold $M \in \mathbb{N}_+$, in order to prevent prolonged computation in case the algorithm fails to converge to a solution of the desired ε-rank. The algorithm for the log-det heuristic is summarized in Algorithm 1, which corresponds to the implemented algorithm addressing ε-rank.

To summarize, the log-det heuristic (Algorithm 1) is guaranteed to find an ε-rank-k solution of the rank-constrained feasibility problem (36) for some $\varepsilon > 0$. If the ε-rank of a solution X_t is equal to k with respect to the chosen value of ε, we have a feasible solution $\hat{X}$ for the rank-constrained optimization problem (30). However, it is possible that the algorithm may not converge to a rank-k solution with respect to the chosen ε, resulting in a higher ε-rank of the obtained solution X_t. In such cases, we can interpret the solution X_t to have a τ-rank equal to k, where $\tau > \varepsilon$. This can provide additional information retrieved from the algorithm beyond its original version, which can be beneficial for assessing the quality of a found solution and analyzing the performance of an algorithm.

Algorithm 1: The implemented algorithm of the log-det heuristic.

Initialize: $X_0 = I_n$;

Set the desired rank: $k \in \mathbb{N}_+$;

Set tolerance: $\varepsilon > 0$;

Set regularization constant: $\delta > 0$;

Set counter of iterations: $t = 0$;

Set counter of the "constant-rank" iterations: $s = 0$;

Set maximum number of the "constant-rank" iterations: $M \in \mathbb{N}_+$;

while ε-$rank(X_t) > k$ *and* $s < M$ **do**

1. set $t = t + 1$;
2. find X_t as a solution of (43);
3. if ε-$rank(X_t) = \varepsilon$-$rank(X_{t-1})$ set $s = s + 1$, otherwise $s = 1$;

end

Output: X_t;

3.2.3 Rank reduction algorithm

This section presents a rank reduction algorithm for solving the rank-constrained feasibility problem (36). This algorithm, proposed in [68], is specifically designed for rank-constrained SDP problems and assumes that $\mathcal{C}$ is an affine set, that is,

$$\mathcal{C} = \{X \mid tr(A_i X) = b_i,\ \forall i = 1, ..., m\}, \tag{44}$$

where $A_1, ..., A_m \in \mathbb{S}^n$ and $b \in \mathbb{R}^m$ are the given data.

As stated earlier, the solution of the semidefinite relaxation (34) is usually a high-rank solution. Therefore, it is reasonable to use rank reduction algorithms to find a solution of the rank-constrained SDP problem (30) when the semidefinite relaxation (34) fails to yield a satisfactory solution. These algorithms aim to iteratively reduce the rank of the initial solution in each iteration. However, the effectiveness of rank reduction is bounded by a specific limit known as the Pataki-Barvinok upper bound on rank, as discussed in [27, §2.9.3]. This upper bound provides an estimation of the maximum rank of a symmetric positive semidefinite matrix, belonging to the feasibility set of the standard SDP problem (12) (12). If this feasibility set (given by m linear equality constraints and a semidefinite

constraint) is non-empty, then there exists a feasible matrix X such that

$$\frac{rank(X)(rank(X)+1)}{2} \leq m, \tag{45}$$

whence the upper bound on rank is

$$rank(X) \leq \lfloor \frac{\sqrt{8m+1}-1}{2} \rfloor. \tag{46}$$

Equivalently we can say that the desired rank k is guaranteed when the number of constraints m satisfies

$$\frac{k(k+1)}{2} \leq m. \tag{47}$$

Although introduced in [68] to find a rank-k solution of the semidefinite relaxation (34), let us first describe the rank reduction algorithm applied to solve the rank-constrained feasibility problem (36) with an affine $\mathcal{C}$ (44), that is, a rank-constrained SDP feasibility problem.

Given a solution $X \in \mathbb{S}_+^n$ of the semidefinite relaxation (34) of the rank-constrained SDP problem (30), the task is to find a solution X^+ of the semidefinite relaxation (34) such that

$$rank(X^+) < rank(X), \tag{48}$$

or equivalently, their nullspaces satisfy $\mathcal{N}(X^+) \supset \mathcal{N}(X)$. This rank reduction algorithm guarantees to find a solution with a rank that is guaranteed by the upper bound on rank (46).

It is known that for any $X \in \mathbb{S}_+^n$, it holds that

$$rank(X) = r \Leftrightarrow X = VV^T, \text{where } V \in \mathbb{R}^{n \times r}. \tag{49}$$

The decomposition (49) can be obtained from the spectral decomposition of X as outlined in Appendix A.3.

If we search for the matrix X^+ in the form

$$X^+ = V(I_r + \alpha\Delta)V^T, \tag{50}$$

we can interpret (50) as a shift of the matrix X

$$X^+ = X + \alpha V \Delta V^T, \tag{51}$$

where $\alpha \in \mathbb{R}$ is the step size, and $\Delta \in \mathbb{S}^r$ is referred to as a direction matrix, as it defines the direction of the shift ([68, §2.2]).

Our goal is to choose $\alpha \in \mathbb{R}$ and $\Delta \in \mathbb{S}^r$ such that X^+ is a solution of the semidefinite relaxation (34) and satisfies (48). It is obvious that α cannot be zero in order to ensure $X^+ \neq X$. In order to maintain feasibility, X^+ needs to satisfy equality constraints of $\mathcal{C}$ (44), which leads to the following conditions for Δ:

$$tr(V^T A_i V \Delta) = 0,\ i = 1, ..., m, \tag{52}$$

where A_i are the matrices defining the equality constraints of $\mathcal{C}$. Furthermore, in order for X^+ to remain positive semidefinite, the following condition must be fulfilled:

$$I_r + \alpha\Delta \succeq 0. \tag{53}$$

To guarantee (48), the matrix $I_r + \alpha\Delta$ has to be singular. To achieve this, we can set $\alpha = -\frac{1}{\lambda_1}$, where λ_1 is the maximum-magnitude eigenvalue of Δ. Consequently, the shifted X^+ defined by (50) will have at least one additional zero eigenvalue, resulting in (48). The procedure is repeated until the desired ε-rank (see Definition 3.1) is obtained. For numerical computation, we can add an additional stopping criterion that limits the number of consecutive iterations providing a solution of the same rank, since the choice of α guarantees that the rank of X decreases at each iteration, only up to the upper bound on rank (46). The algorithm is summarized as Algorithm 2.

To summarize, Algorithm 2 finds a feasible rank-k solution of the rank-constrained optimization problem (30) with respect to a chosen tolerance ε. When the desired rank k satisfies (47), then the stopping criterion that limits the number of consecutive iterations handling a solution of the unchanged ε-rank is unnecessary and the algorithm converges as proved in [68].

As mentioned in the introduction to this algorithm, it was originally designed to solve the rank-constrained SDP problem (30). The idea is to search for a rank-k solution among optimal solutions of the semidefinite relaxation (34). However, it operates under a strong assumption, which is summarized as Assumption 3.1.

Algorithm 2: The rank reduction algorithm for SDP feasibility problems.

Input: X_0 as a solution of the semidefinite relaxation (34);

Set the desired rank: $k \in \mathbb{N}_+$;

Set tolerance: $\varepsilon > 0$;

Set counter of iterations: $t = 0$;

Set counter of the "constant-rank" iterations: $s = 0$;

Set maximum number of the "constant-rank" iterations: $M \in \mathbb{N}_+$;

while ε-$rank(X_t) > k$ *and* $s < M$ **do**

1. define $r = \varepsilon$-$rank(X_t)$;
2. find $V \in \mathbb{R}^{n \times r}$ such that $X = VV^T$;
3. solve a feasibility problem with constraint (52) to find a nonzero Δ;
4. find λ_1 as the maximum-magnitude eigenvalue of Δ;
5. take $\alpha = -\frac{1}{\lambda_1}$;
6. set $t = t + 1$;
7. define $X_t = V(I_r + \alpha\Delta)V^T$;
8. if ε-$rank(X_t) = \varepsilon$-$rank(X_{t-1})$ set $s = s + 1$, otherwise $s = 1$;

end

Output: X_t;

Assumption 3.1. *Let $\mathcal{F}_0^*$ be a set of optimal solutions of the convex relaxation* (34) *of the rank-constrained optimization problem* (30)*. Then there exists a rank-k solution $\hat{X} \in \mathcal{F}_0^*$ that is also an optimal solution of the rank-constrained optimization problem* (30)*.*

In the original version of Algorithm 2, in order to maintain $g(X^+) = g(X)$, that is, $tr(CX^+) = tr(CX)$, it is required that $\Delta \in \mathbb{S}^r$ meets the condition

$$tr(V^T C V \Delta) = 0, \tag{54}$$

where $V \in \mathbb{R}^{n \times r}$ comes from the decomposition (49). Under Assumption 3.1, Algorithm 2 can even find a solution of the rank-constrained SDP problem (30), if the fourth point of the while-loop in Algorithm 2 finds Δ satisfying both (54) and (52), instead of just (52).

Note that Algorithm 2 requires converting any given rank-constrained problem into the standard form of SDP programs (12) with an additional rank constraint. However,

this conversion may be impractical in real-life problems, as observed in Chapter 4 and Chapter 5 when dealing with applications.

3.2.4 Convex iteration as a rank reduction algorithm

In [27, §4.4.2], the author proposed a solution approach for the rank-constrained feasibility problem (36) by iteratively solving two convex problems until convergence. The following sequence of convex problems is solved during the t-th iteration:

$$\begin{aligned} X_t = \quad & \underset{X \in \mathbb{S}^n_+}{\operatorname{argmin}} \quad tr(U_{t-1}X) \\ & X \quad \in \quad \mathcal{C}, \end{aligned} \tag{55}$$

and

$$\begin{aligned} U_t = \quad & \underset{U \in \mathbb{S}^n_+}{\operatorname{argmin}} \qquad tr(UX_t) \\ & I_n - U \succeq 0, \\ & tr(U) = n - k, \end{aligned} \tag{56}$$

where $\mathcal{C}$ is a convex set, $k \in \mathbb{N}_+$ is the desired rank and U_{t-1} in (55) is the so-called direction matrix found by solving (56) in the previous iteration. We set $U_0 = 0$ so that in the first iteration, the problem (55) becomes equivalent to the convex relaxation (34) of the rank-constrained optimization problem (30). On the other hand, if we set $U_0 = I_n$, the first iteration is equivalent to the trace heuristic (39).

We provide insight into how this method works. Due to the optimality of X_t for (55) and the optimality of U_t for (56) in the t-th iteration, we have

$$tr(X_1U_1) \geq tr(X_2U_1) \geq tr(X_2U_2) \geq tr(X_3U_2) \geq tr(X_3U_3) \geq ... \tag{57}$$

Consequently, among iterations, the optimal objective values of the problem (56) satisfy

$$tr(X_1U_1) \geq tr(X_2U_2) \geq tr(X_3U_3) \geq ... \tag{58}$$

Since the objective function of the SDP problem (56) represents the sum of $n-k$ smallest eigenvalues of X_t as introduced in [2] (see Appendix B.2), the iterative process guarantees that for eigenvalues $\lambda_1 \geq ... \geq \lambda_n \geq 0$ of particular solutions $X_1, X_2, X_3, ...$ it holds

$$\sum_{j=k+1}^{n} \lambda_j(X_1) \geq \sum_{j=k+1}^{n} \lambda_j(X_2) \geq \sum_{j=k+1}^{n} \lambda_j(X_3) \geq ... \tag{59}$$

Keeping in mind the statement of Lemma 3.1, the natural effort is to achieve the matrix X_t (for some t) with a zero-sum of its $n-k$ smallest eigenvalues, that is,

$$tr(U_t X_t) = \sum_{j=k+1}^{n} \lambda_j(X_t) = 0. \tag{60}$$

The stopping criterion (60) guarantees that the rank of the matrix X_t is at most k. Theoretically, it does not have to be reached since the inequalities in (59) are not strict. However, it works well in practice, as demonstrated in [27, §4].

In numerical computation, the task is to find a solution with ε-rank (Definition 3.1) equal to k. To ensure that all the $n-k$ smallest eigenvalues are lower than ε, we can adjust the stopping criterion (60) as follows

$$tr(U_t X_t) < \frac{\varepsilon}{n-k}. \tag{61}$$

Since convergence is not always guaranteed, we can also add a stopping criterion that limits the maximum number of consecutive iterations handling a solution of the same rank. The updated algorithm is summarized below as Algorithm 3.

Algorithm 3: The implemented algorithm of the convex iteration.

Initialize: $U_0 = 0$;

Set the desired rank: $k \in \mathbb{N}_+$;

Set tolerance: $\varepsilon > 0$;

Set counter of iterations: $t = 1$;

Set counter of the "constant-rank" iterations: $s = 0$;

Set maximum number of the "constant-rank" iterations: $M \in \mathbb{N}_+$;

Find X_t as a solution of (55);

Find U_t as a solution of (56);

while $tr(X_t U_t) > \frac{\varepsilon}{n-k}$ *and* $s < M$ **do**

1. set $t = t + 1$;
2. find X_t as a solution of (55);
3. find U_t as a solution of (56);
4. if ε-$rank(X_t) = \varepsilon$-$rank(X_{t-1})$ set $s = s + 1$, otherwise $s = 1$;

end

Output: X_t;

Similarly to the log-det heuristic (Algorithm 1), if the convex iteration (Algorithm 3) finds a rank-k solution, it is a feasible solution of the rank-constrained optimization problem (30). In case the algorithm stops on the criterion $s \geq M$, the resulting X_t can be represented as a solution of rank k with respect to a tolerance $\tau > \varepsilon$.

3.3 Methods for solving rank-constrained optimization problems

In this section, we present revised versions of the methods used to solve the rank-constrained feasibility problem (36), as discussed in the previous section, in order to adapt them for solving the rank-constrained optimization problem (30). Let us recall the formulation of the rank-constrained optimization problem:

$$\begin{aligned} \underset{X \in \mathbb{S}^n_+}{min} \quad & g(X) \coloneqq tr(CX) \\ & X \in \mathcal{C}, \\ & rank(X) \leq k, \end{aligned} \tag{30}$$

where $C \in \mathbb{S}^n$, $\mathcal{C}$ is a convex set and $k \in \mathbb{N}_+$ is the desired rank. Let us denote the optimal solution of (30) as X^* with the corresponding optimal value $g^* \coloneqq g(X^*)$.

Furthermore, we propose a bisection algorithm that can find an optimal solution of the rank-constrained optimization problem (30) under some specific assumptions when provided with a feasible solution of the rank-constrained feasibility problem (36).

3.3.1 Bi-criterion heuristics

Standardly, the rank-constraint in the rank-constrained optimization problem (30) is addressed in a similar manner as the rank-constrained feasibility problem (36), where the rank is minimized instead of being constrained. Since the objective function $g(X) \coloneqq tr(CX)$ is minimized in the rank-constrained optimization problem (30), it can be reformulated as a bi-criterion optimization problem of the form

$$\begin{aligned} \underset{X \in \mathbb{S}^n_+}{min} \quad & \min\{g(X), rank(X)\} \\ & X \in \mathcal{C}, \end{aligned} \tag{62}$$

where $\mathcal{C}$ is the convex set of the rank-constrained optimization problem (30).

When solving the bi-criterion problem (62) using scalarization techniques (as described in [19, §4.7.5]), a relative weight $\alpha > 0$ is introduced, resulting in a classic optimization problem of the form

$$\begin{array}{ll} \underset{X \in \mathbb{S}_+^n}{min} & g(X) + \alpha \, rank(X) \\ & X \in \mathcal{C}. \end{array} \tag{63}$$

The methods presented for solving the rank-constrained feasibility problem (36) in the previous section can also be applied to address the rank minimization problem of the form (63), as proposed in [27, 36]. However, these methods involve solving bi-criterion problems, which can be formulated as scalarization problems with the following formulations:

- the bi-criterion version of the trace heuristic (39):

$$\begin{array}{ll} \underset{X \in \mathbb{S}_+^n}{\min} & g(X) + \alpha \, tr(X) \\ & X \in \mathcal{C}, \end{array} \tag{64}$$

  where $\alpha > 0$ is the relative weight.

- the bi-criterion version of the log-det heuristic (43) in the t-th iteration:

$$\begin{array}{ll} X_t = \underset{X \in \mathbb{S}_+^n}{\operatorname{argmin}} & g(X) + \alpha \, tr((X_{t-1} + \delta I_n)^{-1} X) \\ & X \in \mathcal{C}, \end{array} \tag{65}$$

  where $\alpha > 0$ is the relative weight and $\delta > 0$ is a small regularization constant.

- the bi-criterion version of the convex iteration (55) in the t-th iteration:

$$\begin{array}{ll} X_t = \underset{X \in \mathbb{S}_+^n}{\operatorname{argmin}} & g(X) + \alpha \, tr(U_{t-1} X) \\ & X \in \mathcal{C}, \end{array} \tag{66}$$

  where $\alpha > 0$ is the relative weight and U_{t-1} is obtained by (56).

Although the bi-criterion versions of the methods (64), (65), (66) are commonly used to solve the rank-constrained optimization problem (30), they still provide only its feasible solutions. It is worth noting that for $\alpha = 0$, these bi-criterion versions of the methods are equivalent to the convex relaxation (34). Therefore, it is assumed that $\alpha > 0$.

By choosing an appropriate value of $\alpha > 0$, the bi-criterion versions of the methods can yield a rank-k solution $\hat{X}$ with the optimal value $\hat{g} \coloneqq g(\hat{X})$. The feasibility of such $\hat{X}$ for the rank-constrained optimization problem (30) provides an upper bound for the optimal value of (30), which can be expressed as

$$g^* \leq \hat{g}. \tag{67}$$

As the bi-criterion versions of the methods take into account the objective function $g(X)$ of the rank-constrained optimization problem (30), they may offer a better upper bound $\hat{g}$ compared to the original versions of these methods (see (37)), when an appropriate $\alpha > 0$ is chosen. However, using these methods requires dealing with the optimal choice of the relative weight $\alpha > 0$, which can potentially prolong the computation.

3.3.2 Modified heuristics

An alternative perspective on the rank-constrained optimization problem (30) to consider the trade-off between the objective function $g(X)$ and the rank of X, as proposed in [36]. The trade-off graph for the rank-constrained optimization problem (30) can be obtained by solving a modified rank-constrained feasibility problem of the form

$$\begin{array}{lrcl} \text{find} & X \in \mathbb{S}^n_+ & & \\ & X & \in & \mathcal{C}, \\ & g(X) & \leq & \gamma, \\ & rank(X) & \leq & k \end{array} \tag{68}$$

for various values of $\gamma \in \mathbb{R}$ and $k \in \mathbb{N}_+$. Such a trade-off graph is illustrated by Figure 5 and Figure 6, from which an optimal solution X^* can be read.

As the objective function $g(X) = tr(CX)$ is linear, the constraint $g(X) \leq \gamma$ in (68) is also linear. Therefore, the problem (68) fits the structure of the rank-constrained feasibility problem (36) and can be solved using methods from Section 3.2. Specifically, when applying these methods to solve the modified rank-constrained feasibility problem (68), they have the following formulations:

- the modified version of the trace heuristic (39):

$$\begin{aligned} \underset{X \in \mathbb{S}_+^n}{min} \quad & tr(X) \\ X \quad & \in \quad \mathcal{C}, \\ g(X) \quad & \leq \quad \gamma, \end{aligned} \tag{69}$$

  where $\gamma \in \mathbb{R}$ and $\mathcal{C}$ is a convex set from (68).

- the modified version of the log-det heuristic (43) in the t-th iteration:

$$\begin{aligned} X_t = \underset{X \in \mathbb{S}_+^n}{\text{argmin}} \quad & tr((X_{t-1} + \delta I_n)^{-1} X) \\ X \quad & \in \quad \mathcal{C}, \\ g(X) \quad & \leq \quad \gamma, \end{aligned} \tag{70}$$

  where $\gamma \in \mathbb{R}$, $\delta > 0$ is a small regularization constant and $\mathcal{C}$ is a convex set from (68).

- the modified version of the convex iteration (55), (56) in the t-th iteration:

$$\begin{aligned} X_t = \quad \underset{X \in \mathbb{S}_+^n}{\text{argmin}} \quad & tr(U_{t-1} X) \\ X \quad & \in \quad \mathcal{C}, \\ g(X) \quad & \leq \quad \gamma, \end{aligned} \tag{71}$$

  where $\gamma \in \mathbb{R}$, U_{t-1} is obtained by (56) and $\mathcal{C}$ is a convex set from (68).

It is impractical to solve the modified trace heuristic (69), the modified log-det heuristic (70), and the modified convex iteration (71), (56) for every possible value of $\gamma \in \mathbb{R}$. Therefore, our goal is to design a bisection algorithm that selects specific values of γ for which the modified rank-constrained feasibility problem (68) is solved, and finds a solution to the rank-constrained optimization problem (30) within a fixed number of iterations. This algorithm is presented in Subsection 3.3.4.

3.3.3 Low-rank solutions of the convex relaxation

As mentioned in Subsection 3.2.3, the rank reduction algorithm (Algorithm 2) has been designed to search for a low-rank solution within the set of optimal solutions of the semidefinite relaxation (34). This algorithm is applicable when $\mathcal{C}$ is affine (44) and Assumption 3.1 is satisfied, as illustrated in Figure 5. The trade-off graph shows X_0 as an

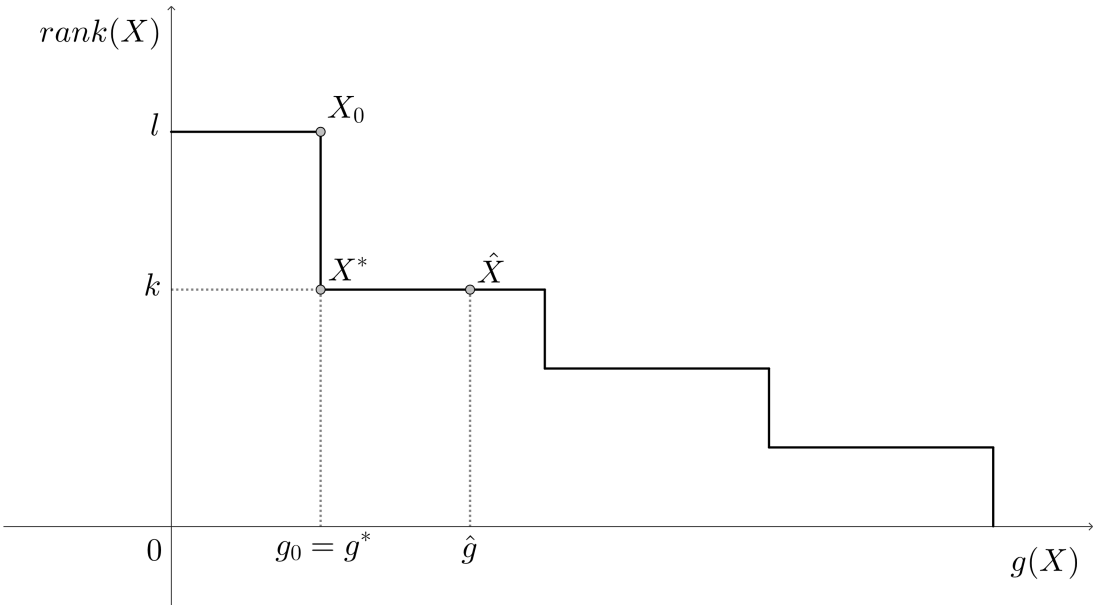


Figure 5: Trade-off between the objective function $g(X)$ and $rank(X)$. Illustration of finding a rank-k solution among the optimal solutions of the relaxation, where X_0 represents a solution of the relaxation with the optimal value $g_0 := g(X_0)$, while X^* denotes an optimal solution of the rank-constrained problem (30) with objective value $g^* := g(X^*)$.

optimal solution of the relaxation (34) with the optimal value $g_0 := g(X_0)$ and it can be observed that there exists a rank-k solution X^* with the objective value $g^* := g(X^*)$ equal to g_0. Since X^* is a feasible solution of the convex relaxation (34) and g_0 gives a lower bound on g^* (see (35)), it follows that X^* is an optimal solution of the rank-constrained optimization problem (30).

Besides using the rank reduction algorithm (Algorithm 2 with (54)) to search for low-rank solutions among solutions of the SDP relaxation (34), we suggest applying the modified trace heuristic (69), the modified log-det heuristic (70) and the modified convex iteration (71), (56) for $\gamma = g_0$. These modified versions of methods for solving the rank-constrained feasibility problem (36) allow searching for low-rank solutions among optimal solution of the convex relaxation (34), not only semidefinite relaxation in particular. Moreover, also in the semidefinite case, they do not require the rank-constrained optimization problem (30) to be given in its standard form of SDP program (12) with additional rank constraint.

The situation where Assumption 3.1 does not hold is depicted in Figure 6 as the opposite scenario. It can be observed that there is no rank-k solution that yields an

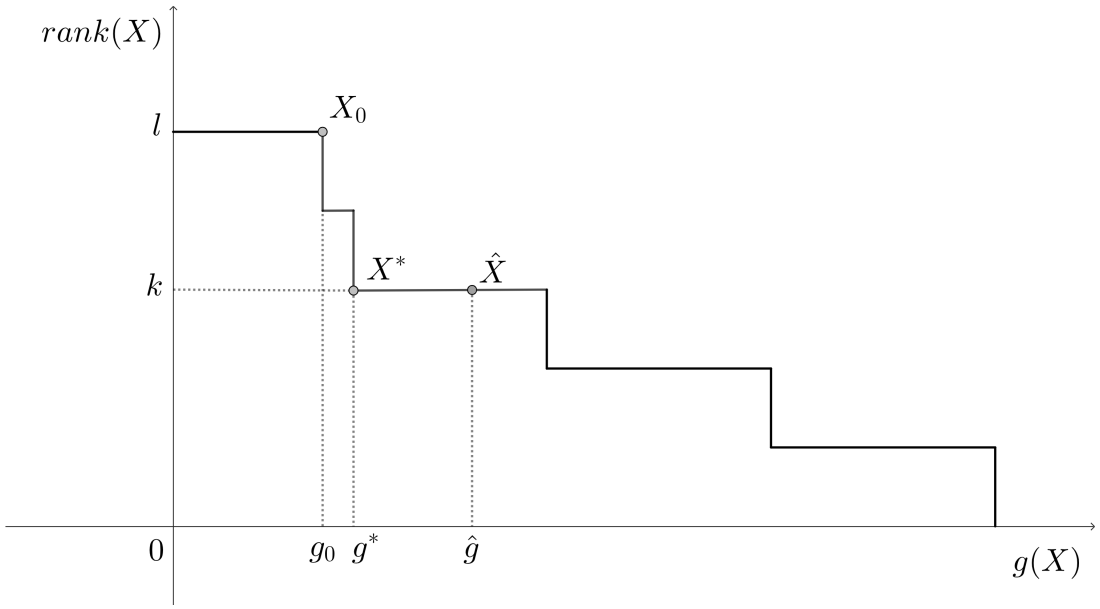


Figure 6: Trade-off between the objective function $g(X)$ and $rank(X)$. Illustration of finding a rank-k solution with the optimal value g^* lying between g_0 and $\hat{g}$, where X_0 represents a solution of the relaxation with the optimal value $g_0 \coloneqq g(X_0)$, X^* denotes an optimal solution of the rank-constrained problem (30) with objective value $g^* \coloneqq g(X^*)$, and $\hat{X}$ represents a feasible solution of the rank-constrained optimization problem with objective value $\hat{g} \coloneqq g(\hat{X})$.

objective value equal to g_0. We address this situation in the following subsection.

3.3.4 Proposed bisection algorithm

Let X^* be an optimal solution of the given rank-constrained optimization problem (30) with the optimal value $g^* \coloneqq g(X^*)$ and the desired rank $rank(X^*) \leq k$. Similarly, X_0 is a solution of the convex relaxation (34) with the optimal value $g_0 \coloneqq g(X_0)$ and $rank(X) = l > k$ and X_1 is a feasible solution of the rank-constrained optimization problem (30), for which $g_1 \coloneqq g(X_1)$ and $rank(X_1) = k$. Our aim is to design a technique that leverages this information to find an optimal solution of the rank-constrained optimization problem (30).

Firstly, it is important to note that the convex relaxation (34) provides a lower bound on the optimal value of the rank-constrained optimization problem (30), as discussed in Section 3.1. This relationship is expressed as

$$g^* \geq g_0. \tag{35}$$

Secondly, as mentioned in Section 3.2, the rank-constrained feasibility problem (36) gives

an upper bound on the optimal value of the rank-constrained optimization problem (30). Recall that we have

$$g^* \leq g_1. \tag{37}$$

Therefore, we can conclude that the optimal value of the rank-constrained optimization problem (30) satisfies

$$g_0 \leq g^* \leq g_1. \tag{72}$$

In the following, we propose a bisection algorithm, which relies on:

i) an optimal solution X_0 of the convex relaxation (34) of the rank-constrained optimization problem (30) with optimal value g_0,

ii) a feasible solution X_1 of the rank-constrained optimization problem (30) with optimal value g_1,

iii) the formulation of the modified rank-constrained optimization problem (68) that allows for finding a solution providing a desired value of the objective of (30),

iv) methods for solving the rank-constrained feasibility problem (36) that can provide either a solution of the required rank or reliable information about its nonexistence.

The algorithm is illustrated in Figure 7, summarized as Algorithm 4, and its properties are formulated in Proposition 3.1.

Proposition 3.1. *Given an interval $[l_1, u_1]$ containing g^* and a small constant $\rho > 0$, the solution $\hat{X}$ provided by Algorithm 4 satisfies*

$$|g^* - \hat{g}| < \rho. \tag{73}$$

Furthermore, Algorithm 4 is guaranteed to find a solution in N iterations, where

$$N = \left\lceil \log_2 \left(\frac{u_1 - l_1}{\rho} \right) \right\rceil. \tag{74}$$

Algorithm 4: Bisection algorithm for solving rank-constrained problems

Input: $g_0 = g(X_0)$ where X_0 is a solution of the convex relaxation (34);

$g_1 = g(X_1)$ where X_1 is a feasible solution of (30);

Initialize: $\hat{X} = X_1$;

Set counter: $t = 1$;

Set tolerance for optimal value: $\rho > 0$;

Set tolerance for rank: $\varepsilon > 0$;

Set interval for γ: $l_t = g_0$, $u_t = g_1$;

while $|u_t - l_t| \geq \rho$ **do**

1. set $\gamma = \frac{l_t + u_t}{2}$;
2. set $t = t + 1$;
3. solve the modified rank-constrained feasibility problem (68) for γ;

if *there exist a solution* X_t *of* (68) **then**

set $\hat{X} = X_t$, $l_t = l_{t-1}$ and $u_t = \gamma$;

else

set set $X_t = X_{t-1}$, $l_t = \gamma$ and $u_t = u_{t-1}$;

end

end

Output: $\hat{X}$ as a ρ-optimal solution of (30) satisfying $rank(\hat{X}) \leq k$, $\hat{g} \coloneqq g(\hat{X})$;

Proof. Denote

$$\mathcal{F} = \{X \in \mathcal{C} \mid g(X) \leq \gamma,\ rank(X) \leq k\}$$

the set of feasible solutions of (68). In each iteration of Algorithm 4 we either find $X_t \in \mathcal{F}$ satisfying $g^* \leq g(X_t) \leq \gamma$ or we find out that no such solution exists. In the latter case we have that for all $X \in \mathcal{F}$ it holds $g(X) > \gamma$ and therefore $g^* = \inf_{X \in \mathcal{F}} \geq \gamma$. Therefore, in each iteration, the property $g^* \in [l_t, u_t]$ is satisfied. Our aim now is to show that in each iteration it holds $g(\hat{X}) \in [l_t, u_t]$. Since, at the initialization, $\hat{X} = X_1$ is chosen so that $g(X_1) = u$, the property is satisfied in the first iteration. Next we show that if $g(X_t) \in [l_t, u_t]$, then $g(X_{t+1}) \in [l_{t+1}, u_{t+1}]$. If (68) is feasible, then $g(X_{t+1}) \leq \gamma = u_{t+1}$. Also, in this case $l_{t+1} = l_t \leq g^* \leq g(X_{t+1})$. On the other hand, if (68) is infeasible, we have that $X_{t+1} = X_t$ and $l_{t+1} = \gamma \leq g^* \leq g(X_t) \leq u_t$. Let $[l_N, u_N]$ be the final interval satisfying $u_N - l_N < \rho$. We have that the both values g^* and $g(\hat{X})$ belong to $[l_N, u_N]$ and

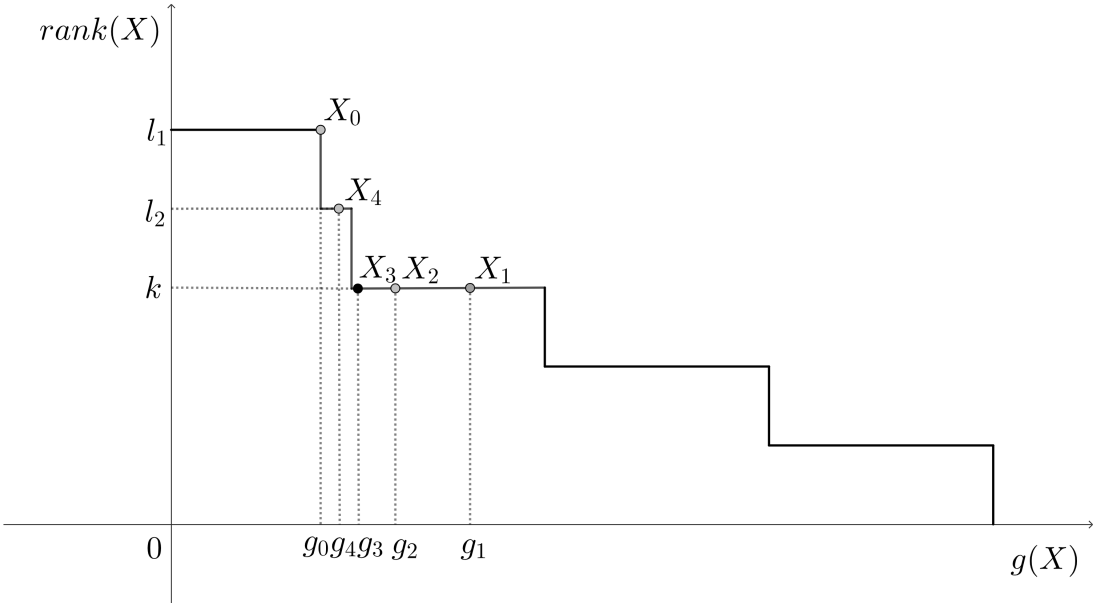


Figure 7: Illustration of the bisection algorithm (Algorithm 4). Starting with the interval $[g_0, g_1]$ provided by a solution X_0 of the relaxation (34) and a feasible solution X_1 of the rank-constrained problem (30). In the first iteration, X_2 denotes a feasible solution with objective value g_2. Since $rank(X_2) = k$, the interval is reduced to $[g_0, g_2]$. The procedure is repeated until the interval $[g_4, g_3]$ is obtained having a length below ε. Finally, the solution X_3 is marked as an optimal solution, as it provides the lowest value of $g(X)$ among the considered feasible rank-k solutions.

therefore

$$|g(\hat{X}) - g^*| \leq u_N - l_N < \rho.$$

In addition, the initial interval $[l_1, u_1]$ is reduced until

$$\frac{u_1 - l_1}{2^t} < \rho \tag{75}$$

for some t. From here, we can conclude that for a fixed $\rho > 0$, the number of iterations N is given by (74).

□

3.3.5 Computational aspects of the bisection algorithm

In this subsection, we discuss factors that affect how well the algorithm works, such as the constant ρ that determines the accuracy of the final solution $\hat{X}$. We will examine the computational issues that could arise if the condition stated in iv) is not met. It is

worth noting that this is precisely the situation we face since only methods described in Section 3.2 are at our disposal to solve the problem (68). In addition, we have to deal with the ε-rank in the implementation of the algorithm.

Since the number of iterations in Algorithm 4 depends on the length of the initial interval $[l_1, u_1]$ and the chosen $\rho > 0$ (see (75)), it can be useful to accelerate the computation by finding a tighter initial interval $[l_1, u_1]$. Rather than loosening the accuracy tolerance ρ, one can find a lower value of the upper bound $u_1 = g_1$, for example, by solving the bi-criterion algorithms of Section 3.2, namely the bi-criterion trace heuristic (64), the bi-criterion log-det heuristic (65) or the bi-criterion convex iteration (66), (56). Since these methods also take into account the minimization of the original objective function $g(X)$, they could give a better upper bound on g^*. For this purpose, the choice of the relative weight $\alpha > 0$ can be arbitrary so that we have a feasible solution of the desired ε-rank. We analyze this hypothesis when solving generated problems in Section 4.4.5.

As we discussed in previous subsections, there are methods for solving the rank-constrained feasibility problem (36), such as the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3). However, these are not guaranteed to find a feasible solution of (30) even if such a solution exists. Therefore, if these algorithms are used to solve (68) in step 3 of Algorithm 4, they can provide a solution with ε-rank higher than k even for $\gamma \geq g^*$. As a result, the found solution $\hat{X}$ can only be considered as an approximation of the optimal solution X^*. On the other hand, when changing the value of ε, the algorithm can find another approximation of the optimal solution X^*. Nevertheless, even the bisection algorithm (Algorithm 4) implemented with Algorithm 1 or Algorithm 3 can work in practice, as we observe in numerical experiments in the upcoming sections.

4 Correlation matrix approximation

This chapter focuses on finding the optimal approximation of an empirical correlation matrix, which is a symmetric matrix with multiple negative eigenvalues. In the literature, this problem is known as the nearest correlation matrix problem, as discussed in [60] and [57]. Throughout this chapter, we will refer to it as the "NCM problem". Additionally, we will discuss the "rank-constrained NCM problem", which involves finding the closest low-rank correlation matrix to a given empirical correlation matrix.

Recall the formulations of the NCM problem and the rank-constrained NCM problem from Example 1.1. For a given empirical correlation matrix $C \in \mathbb{S}^n$, the NCM problem is formulated as follows

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & \|C - X\|_F \\ & X_{ii} = 1, \quad i = 1, ..., n, \\ & X \succeq 0. \end{aligned} \tag{3}$$

When the approximation of the given empirical correlation matrix $C \in \mathbb{S}^n$ is assigned to have a desired rank $k \in \mathbb{N}_+$, the rank-constrained NCM problem is formulated in the form

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & \|C - X\|_F \\ & X_{ii} = 1, \quad i = 1, ..., n, \\ & X \succeq 0, \\ & rank(X) \leq k. \end{aligned} \tag{4}$$

Note that the NCM problem (3) can be viewed as the rank-constrained NCM problem (4), if we choose $k = n$.

The NCM problem (3) and the rank-constrained NCM problem (4) can be considered as representatives of standard matrix approximation problems, where $m = n$, $A = I_n$ and $B = I_n$ in their generalized formulation (1). The task is to find a (low-rank) matrix $X \in \mathbb{S}^n$ that satisfies the properties of a correlation matrix as defined in Definition 1.1, and also serves as the best approximation of the empirical correlation matrix $C \in \mathbb{S}^n$ under the Frobenius norm.

The NCM problem (3) arises when the data used to construct the correlations are asynchronous or incomplete, or when the models are stress-tested by artificially adjusting

individual correlations. It is crucial to solve the NCM problem to prevent subsequent calculations from breaking down due to negative variances or volatilities, as explained in [60].

As stated in [57] and [106], the rank-constrained NCM problem arose as part of the calibration of the so-called multi-factor market model of interest rates. Financial institutions use this model to price their portfolios of interest rate derivatives, with interest rates as variables assumed to follow log-normal stochastic processes. Due to the use of historical data, a correlation structure of interest rates can be extracted. The idea of the model is then to implant the correlation structure into the stochastic processes for the interest rates so that the model can appropriately describe the dynamics of interest rates and pricing can be more accurate. If the model works with k factors, it is evident that the rank of the correlation matrix should not exceed k. Such a correlation matrix cannot be used if the rank is higher than the number of factors, which is almost always the case. Therefore, a low-rank correlation matrix is required that best approximates the given empirical correlation matrix under the Frobenius norm, which makes the problem of finding the nearest low-rank correlation matrix (4) so significant. It is essential to note that in practical applications the number of interest rates in the model can be enormous, resulting in a high-dimensional correlation matrix. On the other hand, the term structure of interest rates is driven by multiple factors (four or more), but definitely fewer than the dimension of the matrix.

The following subsections provide an overview of publications addressing the (low-rank) correlation matrix approximation. Furthermore, we present our reformulations of these problems using the transformations introduced in Chapter 2. A significant part of this chapter is devoted to discussing numerical results. When solving the generated rank-constrained NCM problems, we focus on demonstrating the performance of the bisection algorithm (Algorithm 4) proposed in Chapter 3. Some of the results summarized in this chapter were previously published in the conference paper [44].

4.1 Literature review

The NCM problem (3) and its rank-constrained version (4) gained attention in the 2000s due to their significance in various fields such as finance, machine learning, and signal

processing. In particular, researchers were interested in developing techniques for estimating the structure of a specific correlation matrix. This section offers a brief outline of the most well-known algorithms for solving these problems.

In the literature, the NCM problem (3) is first mentioned in [60]. As the author explains, he was approached by a London fund management company that encountered this problem in 2000. In [60], an alternating projections algorithm is derived for computing (3), which is an iterative algorithm that projects at each iteration onto the set of matrices with unit diagonal and the cone of symmetric positive semidefinite matrices. Although the alternating projections algorithm is widely used, it can converge slowly, especially for large matrices. In 2006, the authors of [82] derived a preconditioned Newton method to solve the NCM problem, which deals with the unconstrained dual of the original problem and has quadratic convergence. These results were further improved in [16], where an algorithm was introduced that provided a speed-up over the original preconditioned Newton method and was applicable to large-dimensional matrices. Both the alternating projections algorithm and the algorithm introduced in [16] have been implemented in the NAG Library in Matlab [71].

A few years later, the rank-constrained NCM problem (4) was first mentioned in the work [106]. The authors noticed the need to find a low-rank approximation to a correlation matrix in the publication [20] that deals with the market model of interest rate dynamics. They transformed the rank-constrained NCM problem into a minimization-maximization problem using the Lagrange multiplier method, which allowed them to solve the inner problem by spectral decomposition and the outer problem by the gradient descent method. Shortly after, majorization was suggested as a suitable rank reduction method in [81] followed by the introduction of a geometric optimization algorithm based on parameterizing the constraint set by the Cholesky manifold in [57]. This publication also established a connection between the algorithm and the Lagrange multiplier method. Later, in 2010, the authors of [17] tackled the rank-constrained NCM problem and derived an explicit solution for the case when $k = 1$. For the general case, they investigated several numerical methods designed for specific norm minimization problems, including the alternating projections algorithm and the spectral projected gradient method. Although the alternating projections method lacks convergence results, it works well in practice. In

comparison, the projected spectral gradient outperformed other approaches. This method is also included in the NAG Library in Matlab [71].

There have been attempts to formulate the NCM problem as an SDP program. As discussed in [60], using the vectorization of a matrix to deal with the Frobenius norm, as introduced in [94] leads to a large-scale problem that cannot be solved by standard SDP solvers. Later in [7], an SDP approach was introduced to solve the NCM problem. In reality, they handle a mixed-cone formulation, for which they derive specific primal-dual interior-exterior-point methods. The authors of [69] also used SDP to represent the sum of the k largest eigenvalues (see Appendix B.3) and introduced an iterative method called the semi-smooth Newton method. From this point of view, our SDP reformulation gives a new insight into the NCM problem. Today, the rank-constrained NCM problem (4) remains an active area of research, as evidenced by recent publications such as [31] and [93].

4.2 SDP reformulation of the NCM problem

In this section, we apply the transformations from Section 2.3 to convert the NCM problem (3) into an SDP program of the form (12). It should be noted that [60] considers an SDP reformulation of the NCM problem (3) to be impractical to handle the Frobenius norm in the objective due to the enormous number of constraints arising after using the vectorization of matrix $C - X$. Therefore, we propose a new approach to handle the Frobenius norm that allows us to solve the SDP program efficiently using standard solvers.

As the feasibility set of the NCM problem (3) is defined by linear and semidefinite constraints, our focus is solely on handling the Frobenius norm in the objective. Using the result of Proposition 2.5, we can equivalently express the NCM problem in the form

$$\begin{aligned} \min_{X\in\mathbb{S}^n, Z\in\mathbb{S}^n} \quad & tr(Z) \\ & X_{ii} = 1, \quad i = 1, ..., n, \\ & X \succeq 0, \\ & Z \succeq (C - X)(C - X)^T. \end{aligned} \tag{76}$$

To sum up the transformation, a new symmetric variable $Z \in \mathbb{S}^n$ was introduced and the Frobenius norm in the objective was replaced by the linear function $tr(Z)$ and an additional constraint $Z \succeq (C - X)(C - X)^T$.

Using the Schur complement property from Lemma 2.1, the last constraint of (76) can be expressed as a semidefinite constraint of the form

$$\begin{bmatrix} I_n & (C-X)^T \\ C-X & Z \end{bmatrix} \succeq 0. \tag{77}$$

Hence, we obtain an equivalent SDP reformulation of the NCM problem (3) that has the following formulation

$$\begin{aligned} \min_{X\in\mathbb{S}^n, Z\in\mathbb{S}^n} \quad & tr(Z) \\ X_{ii} &= 1, \quad i = 1, ..., n, \\ X &\succeq 0, \\ \begin{bmatrix} I_n & (C-X)^T \\ C-X & Z \end{bmatrix} &\succeq 0. \end{aligned} \tag{78}$$

Our SDP reformulation (78) allows us to exclusively use SDP tools to solve the NCM problem (3). Furthermore, our approach is not limited to handling the Frobenius norm, as it can also handle other norms in the objective of the NCM problem, such as the l_1, l_2 or l_∞ norm (see Section 2). Additionally, our approach can handle the problem of completing a correlation matrix with the objective $\|W \circ (C - X)\|_F$, where $W \in \mathbb{S}^n$ is defined as in (2). As the Hadamard product (Definition A.4) in the objective does not affect our SDP reformulation, it only requires reformulating the constraint (77) in the form

$$\begin{bmatrix} I_n & (W \circ (C-X))^T \\ W \circ (C-X) & Z \end{bmatrix} \succeq 0. \tag{79}$$

4.3 SDP reformulation of the rank-constrained NCM problem

When considering the rank-constrained NCM problem (4), we deal with the Frobenius norm in the objective by applying the same procedure as in the NCM problem (3) described earlier. This yields an equivalent rank-constrained SDP problem that takes the

form

$$\begin{aligned} \min_{X\in\mathbb{S}^n, Z\in\mathbb{S}^n} \quad & tr(Z) \\ X_{ii} &= 1, \quad i = 1, ..., n, \\ X &\succeq 0, \\ rank(X) &\leq k, \\ \begin{bmatrix} I_n & (C-X)^T \\ C-X & Z \end{bmatrix} &\succeq 0. \end{aligned} \tag{80}$$

The rank-constrained SDP reformulation (80) of the rank-constrained NCM problem (4) allows the use of methods to solve rank-constrained SDP problems presented in Chapter 3. It is worth noting that the SDP reformulation of the NCM problem (78) is actually the SDP relaxation of the rank-constrained SDP reformulation (80). As explained in Section 3.1, standard solvers for SDP problems are known to converge to a high-rank optimal solution. Therefore, one must consider using rank minimization heuristics or rank reduction algorithms if the solution of the SDP relaxation (78) fails to provide a rank-k solution.

Due to the constraint on the unit diagonal of the matrix variable X, whose rank is restricted, the trace heuristic is equivalent to the SDP relaxation, since the trace of X is a constant function. However, other algorithms in Chapter 3 do not have this drawback. Therefore, we offer their formulations to solve the rank-constrained NCM problem (4) below.

In Algorithm 1, the log-det heuristic (43) applied to the rank-constrained NCM problem (80) has the form

$$\begin{aligned} X_t = \operatorname*{argmin}_{X\in\mathbb{S}^n, Z\in\mathbb{S}^n} \quad & tr((X_t + \delta I_n)^{-1} X) \\ X_{ii} &= 1, \quad i = 1, ..., n, \\ X &\succeq 0, \\ \begin{bmatrix} I_n & (C-X)^T \\ C-X & Z \end{bmatrix} &\succeq 0, \end{aligned} \tag{81}$$

where $C \in \mathbb{S}^n$ is the given empirical correlation matrix and $\delta > 0$ is a small regularization constant.

In Algorithm 3, the convex iteration (55), (56) applied to solve the rank-constrained

NCM problem (80) is formulated as follows

$$\begin{aligned} X_t = \underset{X\in\mathbb{S}^n, Z\in\mathbb{S}^n}{\operatorname{argmin}} \quad & tr(U_{t-1}X) \\ & X_{ii} = 1, \quad i = 1, ..., n, \\ & X \succeq 0, \\ & \begin{bmatrix} I_n & (C-X)^T \\ C-X & Z \end{bmatrix} \succeq 0, \end{aligned} \tag{82}$$

and

$$\begin{aligned} U_t = \underset{U\in\mathbb{S}^n}{\operatorname{argmin}} \quad & tr(X_t U) \\ & 0 \preceq U \preceq I_n, \\ & tr(U) = n - k, \end{aligned} \tag{83}$$

where $C \in \mathbb{S}^n$ is the given empirical correlation matrix and $k \in \mathbb{N}_+$ is the desired rank. The bi-criterion and modified versions of the log-det heuristic and the convex iteration can be formulated analogously to (65), (66), and (70), (71), respectively.

Unlike the algorithms mentioned above, applying the rank reduction algorithm (Algorithm 2) requires a reformulation of the rank-constrained SDP reformulation (80) into the standard form of the SDP problem (12) with an additional rank constraint. To achieve this, we introduce the new variable $Y \in \mathbb{S}^{3n}$, which yields a standard-like reformulation of the rank-constrained NCM problem (4) in the form

$$\begin{aligned} \min_{Y\in\mathbb{S}^{3n}, X\in\mathbb{S}^n} \quad & tr(Y) \\ & Y_{ii} = 1, \;\; i = 1, ..., n, \\ & Y = \begin{bmatrix} X & 0 & 0 \\ 0 & I_n & (C-X)^T \\ 0 & C-X & Z \end{bmatrix} \succeq 0, \\ & rank(Y) \leq n + k. \end{aligned} \tag{84}$$

It should be noted that, apart from the unit diagonal of $X \in \mathbb{S}^n$, there are also numerous linear constraints that define $Y \in \mathbb{S}^{3n}$ as a block matrix containing another matrix variable $X \in \mathbb{S}^n$. Consequently, even if we establish specific requirements for Δ in Algorithm 2, the algorithm is guaranteed to converge only for high values of the desired rank k due to the upper bound (45).

However, we can take advantage of the fact that we modified the original version of Algorithm 2 to find a feasible solution for the rank-constrained problem instead of

seeking a rank-k solution among optimal solutions to the SDP relaxation. Thanks to the semidefinite structure of the original rank-constrained NCM problem (4), we can directly apply Algorithm 2 to find a feasible solution. Note that there are n linear constraints that define the feasibility set of the NCM problem (4), implying that Algorithm 2 is guaranteed to converge to a feasible rank-k solution if $k \leq \lfloor \frac{\sqrt{8n+1}-1}{2} \rfloor$, as shown in (46).

4.4 Numerical results

This section presents the numerical results of the NCM problem (3) and the rank-constrained NCM problem (4). First, we deal with the problems from Example 1.1 to illustrate how the algorithms work. Then, we validate the results of the SDP reformulation (78) while solving the NCM problem (3) using standard algorithms involved in the NAG library in Matlab [71]. Last but not least, we solve the rank-constrained NCM problem (4) to demonstrate the performance of the bisection algorithm (Algorithm 4). We also compare algorithms for solving rank-constrained NCM problems whose formulations are presented in Section 4.3.

Our experiments were carried out in MATLAB R2019a [71] on an Intel Core i7-4690 CPU processor running at 3.6GHz. To solve SDP problems, we used the SDPT3 solver, which is included in the CVX modeling system [56, 55], a package that specifies and solves convex problems. The empirical correlation matrices were generated as symmetric matrices with a unit diagonal, and their elements were drawn from the interval [-1,1]. We computed the rank of a matrix as a ε-rank (see Definition 3.1) for $\varepsilon = 10^{-6}$. In all algorithms, we use these inputs: $\rho = 10^{-6}$, $M = 20$, $\delta = 0.01$ and $\alpha = 10$.

4.4.1 Illustrative example

First, recall the assignment of Example 1.1. Given an empirical correlation matrix

$$C = \begin{pmatrix} 1 & -0.323 & 0.146 & 0.553 & -0.252 & 0.201 & -0.034 & -0.241 \\ -0.323 & 1 & 0.260 & 0.140 & 0.573 & 0.015 & 0.269 & 0.282 \\ 0.146 & 0.260 & 1 & -0.060 & 0.788 & 0.774 & -0.718 & 0.910 \\ 0.553 & 0.140 & -0.060 & 1 & -0.006 & 0.074 & 0.499 & -0.230 \\ -0.252 & 0.573 & 0.788 & -0.006 & 1 & 0.890 & -0.220 & 0.881 \\ 0.201 & 0.015 & 0.774 & 0.074 & 0.890 & 1 & -0.193 & 0.822 \\ -0.034 & 0.269 & -0.718 & 0.499 & -0.220 & -0.193 & 1 & -0.537 \\ -0.241 & 0.282 & 0.910 & -0.230 & 0.881 & 0.822 & -0.537 & 1 \end{pmatrix}$$

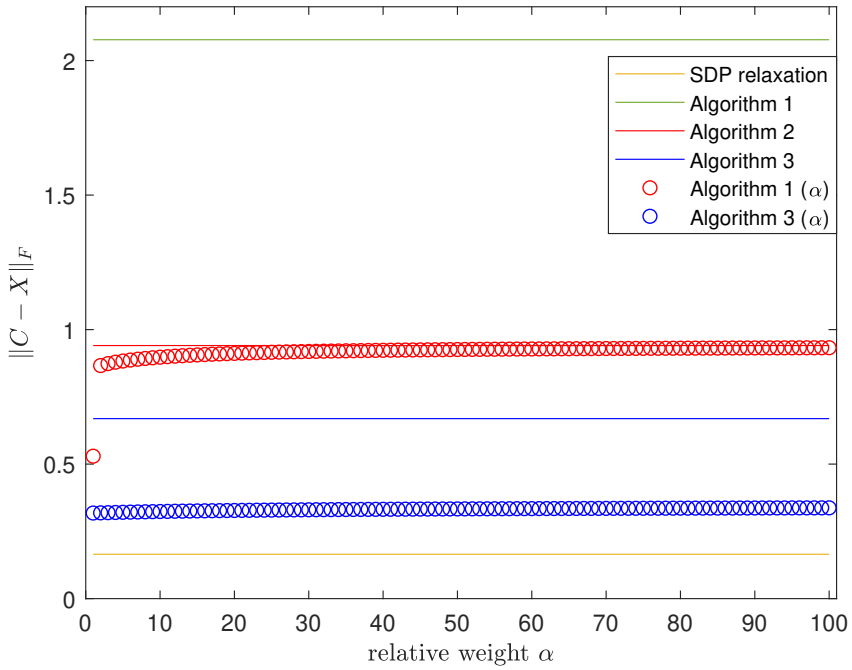


Figure 8: Optimal values yielded by different algorithms in solving Example 1.1. Values of the objective $\|C - X\|_F$, where X is a solution obtained by a particular algorithm, with respect to the relative weight $\alpha > 0$.

with eigenvalues $-0.13, -0.04, 0.05, 0.23, 0.63, 1.63, 1.74, 3.89$, the task is to find its nearest correlation matrix under the Frobenius norm. After solving the SDP reformulation (78) of the NCM problem (3), we obtain an optimal approximation of C in the form

$$X = \begin{pmatrix} 1 & -0.332 & 0.136 & 0.551 & -0.227 & 0.178 & -0.031 & -0.230 \\ -0.332 & 1 & 0.240 & 0.146 & 0.542 & 0.054 & 0.246 & 0.277 \\ 0.136 & 0.240 & 1 & -0.067 & 0.787 & 0.754 & -0.683 & 0.894 \\ 0.551 & 0.146 & -0.067 & 1 & -0.014 & 0.086 & 0.491 & -0.231 \\ -0.227 & 0.542 & 0.787 & -0.014 & 1 & 0.826 & -0.200 & 0.902 \\ 0.178 & 0.054 & 0.754 & 0.086 & 0.826 & 1 & -0.227 & 0.806 \\ -0.031 & 0.246 & -0.683 & 0.491 & -0.200 & -0.227 & 1 & -0.539 \\ -0.230 & 0.277 & 0.894 & -0.231 & 0.902 & 0.806 & -0.539 & 1 \end{pmatrix}$$

with eigenvalues $0, 0, 0.02, 0.22, 0.59, 1.60, 1.73, 3.85$. The computation took 0.006 seconds and the optimal value of $\|C - X\|_F$ is 0.1649. From the viewpoint of these results, the NCM problem (3) can be seen as a tool forcing negative eigenvalues of the empirical correlation matrix to become zeros and correspondingly adjusting particular entries of X (representing correlations) to achieve this.

Although the NCM problem (3) aims to find a correlation matrix with several zero

eigenvalues, the resulting rank of the correlation matrix may not be as low as needed for the application. As can be seen from its eigenvalues, the matrix X in Example 1.1 has a rank of 6. If we consider the rank-constrained NCM problem (4) with $k = 4$, the SDP relaxation (the NCM problem (3)) alone is insufficient to obtain a rank-4 solution. Therefore, we must apply methods for solving rank-constrained SDP problems from Section 3.2 and Section 3.3 to achieve a rank-4 approximation of C.

Figure 8 shows the distance between the empirical correlation matrix C and its approximation X found using different algorithms, with respect to the chosen relative weight $\alpha > 0$ required in the bi-criterion version of the log-det algorithm (Algorithm 1 with (65)) and the bi-criterion version of the convex iteration (Algorithm 1 with (66)). The SDP (78) yields a rank-6 solution that provides a lower bound on the optimal value of the rank-constrained NCM problem (4). We can apply algorithms for finding a feasible solution, that is, the log-det heuristic (Algorithm 1 with (81)), the rank reduction algorithm (Algorithm 2) and the convex iteration (Algorithm 3 with (82), (83)). Since they are not bi-criterion, their values of $\|C - X\|_F$ are displayed as constant functions in Figure 8. We can see that the rank reduction algorithm (Algorithm 2) found a feasible solution that provides the worst approximation of C. Among the feasible solutions found (displayed as lines), the convex iteration (Algorithm 3 with (82), (83)) yields the best approximation of the empirical matrix C.

These approximations can be improved when using the bi-criterion versions of the log-det heuristic (65) and the bi-criterion version of the convex iteration (66), (56). However, Figure 8 indicates that both bi-criterion algorithms yield worse results when using higher values of the relative weight $\alpha > 0$. The bi-criterion version of the log-det heuristic (Algorithm 1 with (65)) only slightly outperforms the original version of the algorithm in terms of objective values. In contrast, the bi-criterion version of the convex iteration (Algorithm 3 with (66), (56)) delivers significantly better approximations of C even for high values of $\alpha > 0$. Notably, a higher value of α implies a smaller weight assigned to minimizing $\|C - X\|_F$. In Section 4.4.5, we will investigate whether such a comparison of algorithms holds in general.

The bisection algorithm (Algorithm 4) can improve the approximations of the empirical correlation matrix C by utilizing the interval $[g_0, g_1]$, which specifies the optimal

value g^* of the rank-constrained NCM problem (4). Here, g_0 is the optimal value of the NCM problem (3), and g_1 is the value provided by a solution obtained by the log-det heuristic (Algorithm 1) or the convex iteration (Algorithm 3). As observed in Figure 8, the bi-criterion versions of the methods lead to a better rank-4 feasible solution X_1 for the bisection algorithm (Algorithm 4).

Table 4 summarizes the results obtained by the bisection algorithm (Algorithm 4). We compare the performance of the algorithm executed with the modified log-det heuristic (70) and the modified convex iteration (71), (56). Since we start with a wider initial interval $[g_0, g_1]$ in the case of the log-det heuristic, the bisection algorithm (Algorithm 4) takes one more iteration to reduce the interval so that it has a length below $\varepsilon = 10^{-6}$. However, the computation time is shorter because only one optimization problem is solved in each iteration, unlike the convex iteration (55), (56), which solves two optimization problems. Note that small deviations between values of $\hat{g}$ and $\|C - \hat{X}\|_F$ obtained by the bisection algorithm (Algorithm 4) using the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3) are caused only by the difference in initial intervals. When calling Algorithm 1 and Algorithm 3, we allowed a maximum of $M = 20$ consecutive iterations to handle a solution with the same ε-rank. It means computation time is prolonged by exactly those iterations of the bisection algorithm (Algorithm 4) where a rank-4 solution is not found. When we set $M = 5$, the bisection algorithm (Algorithm 4) using the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3) took 15.65 seconds, and 45.04 seconds, respectively. These times are significantly lower than those obtained when $M = 20$ and the results remain unchanged. In the last column of Table 4, we present the maximum value of τ, which can be interpreted as the value of ε that would ensure Algorithm 4 finds a solution of ε-rank equal to 4 in each iteration.

Figure 9 displays the first iterations and the final solutions $\hat{X}$ for both versions of the bisection algorithm (Algorithm 4): the log-det version on the left and the convex iteration version on the right. These two graphs differ in the initial interval determined by X_0 and X_1, therefore, while the log-det version finds an optimal solution $\hat{X}$ in the first half of the initial interval, the convex iteration version only slightly improves the initial approximation X_1. A more general comparison of these versions of the bisection algorithm (Algorithm 4) is provided in Section 4.4.4 and Section 4.4.5.

bisection	g_0	g_1	$\hat{g}$	$\|C - \hat{X}\|_F$	time (s)	iter.	max. τ
log-det heu.	0.027192	0.280327	0.101004	0.317811	48.53	19	0.0167
convex iter.	0.027192	0.101204	0.100995	0.317796	109.71	18	0.0657

Table 4: Results obtained by the bisection algorithm in solving Example 1.1. Solving the rank-constrained NCM problem (4) from Example 1.1. Comparison of the results obtained by the bisection algorithm (Algorithm 4) based on solving the modified rank-constrained feasibility problems either by the log-det heuristic (Algorithm 1) or the convex iteration (Algorithm 3) in step 3 of Algorithm 4.

Overall, using the log-det heuristic, Algorithm 4 was able to find a correlation matrix $\hat{X}$ with nonzero eigenvalues of 3.8837, 1.8366, 1.658, and 0.6217. On the other hand, the convex iteration yielded a solution $\hat{X}$ with nonzero eigenvalues of 3.8824, 1.8362, 1.6534, and 0.6241. The difference between these solutions, measured by the Frobenius norm, was 0.0031. Therefore, after rounding the entries of $\hat{X}$, both solutions provided an approximation of C of the form

$$\hat{X} = \begin{pmatrix} 1 & -0.388 & 0.147 & 0.672 & -0.240 & 0.154 & -0.085 & -0.211 \\ -0.388 & 1 & 0.247 & 0.215 & 0.542 & 0.030 & 0.220 & 0.294 \\ 0.147 & 0.247 & 1 & -0.072 & 0.790 & 0.756 & -0.688 & 0.903 \\ 0.672 & 0.215 & -0.072 & 1 & 0 & 0.122 & 0.556 & -0.269 \\ -0.240 & 0.542 & 0.790 & 0 & 1 & 0.828 & -0.199 & 0.924 \\ 0.154 & 0.030 & 0.756 & 0.122 & 0.828 & 1 & -0.235 & 0.832 \\ -0.085 & 0.220 & -0.688 & 0.556 & -0.199 & -0.235 & 1 & -0.535 \\ -0.211 & 0.294 & 0.903 & -0.269 & 0.924 & 0.832 & -0.535 & 1 \end{pmatrix}.$$

In summary, we achieve an approximation of the given empirical correlation matrix that satisfies the properties of a correlation matrix from Definition 1.1 and has the desired rank. The structure of Algorithm 4 ensures that the found approximation is the best possible under the Frobenius norm with respect to the specified tolerance ε. It is important to note that Algorithm 4 is limited by the specifications of iterative algorithms, namely the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3), such as the stopping criterion based on the numerical ε-rank and a maximum of M consecutive iterations in which no change of rank occurred. Despite these limitations, Algorithm 4 succeeded in finding a rank-4 approximation to the given empirical correlation matrix C that is superior to the one obtained by a standard method.

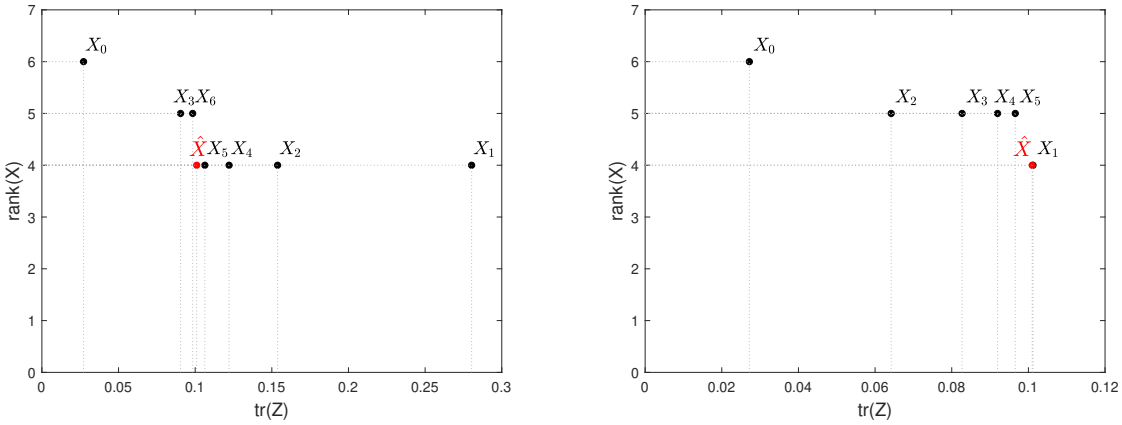


Figure 9: Trade-off between the objective $tr(Z)$ and $rank(X)$. Displayed solutions are obtained by Algorithm 4 while solving the rank-constrained NCM problem (4) from Example 1.1. The graphs illustrate the first iterations of the algorithm using the modified log-det heuristic (left) and the modified convex iteration (right). The final solution $\hat{X}$ is highlighted in red.

Correlation matrix completion

At this place, we also want to demonstrate that the conic optimization approach can be applied to solve matrix completion problems. Suppose that the given empirical correlation matrix C contains some missing entries (NaNs), such as

$$C_{miss} = \begin{pmatrix} 1 & NaN & 0.146 & 0.553 & -0.252 & 0.201 & -0.034 & -0.241 \\ NaN & 1 & 0.260 & 0.140 & 0.573 & 0.015 & 0.269 & 0.282 \\ 0.146 & 0.260 & 1 & -0.060 & NaN & 0.774 & -0.718 & 0.910 \\ 0.553 & 0.140 & -0.060 & 1 & -0.006 & NaN & 0.499 & -0.230 \\ -0.252 & 0.573 & NaN & -0.006 & 1 & 0.890 & -0.220 & 0.881 \\ 0.201 & 0.015 & 0.774 & NaN & 0.890 & 1 & -0.193 & 0.822 \\ -0.034 & 0.269 & -0.718 & 0.499 & -0.220 & -0.193 & 1 & NaN \\ -0.241 & 0.282 & 0.910 & -0.230 & 0.881 & 0.822 & NaN & 1 \end{pmatrix}.$$

In this case, it is not sufficient to solve only the standard form of the NCM problem (3), as we also need to handle the missing entries in C_{miss} in the objective. To address this issue, we introduce a matrix $W \in \mathbb{S}^n$ defined as (2) to solve the problem of finding the completion of the correlation matrix, such that the found approximation X minimizes $\|W \circ (C_{miss} - X)\|_F$. This can be achieved by solving the NCM problem (78) with the objective $\|W \circ (C_{miss} - X)\|_F$ and the updated last constraint in (79). The resulting

completion of the empirical correlation matrix C_{miss} has the following form

$$X = \begin{pmatrix} 1 & -0.481 & 0.128 & 0.553 & -0.238 & 0.191 & -0.039 & -0.231 \\ -0.481 & 1 & 0.233 & 0.140 & 0.547 & 0.054 & 0.245 & 0.277 \\ 0.128 & 0.233 & 1 & -0.060 & 0.789 & 0.752 & -0.683 & 0.897 \\ 0.553 & 0.140 & -0.060 & 1 & -0.006 & 0.141 & 0.499 & -0.230 \\ -0.238 & 0.547 & 0.789 & -0.006 & 1 & 0.836 & -0.204 & 0.901 \\ 0.191 & 0.054 & 0.752 & 0.141 & 0.836 & 1 & -0.225 & 0.803 \\ -0.039 & 0.245 & -0.683 & 0.499 & -0.204 & -0.225 & 1 & -0.580 \\ -0.231 & 0.277 & 0.897 & -0.230 & 0.901 & 0.803 & -0.580 & 1 \end{pmatrix}$$

providing the objective value equal to 0.1487 and having eigenvalues of 0.1597, 0.5324, 1.6813, 1.7658, and 3.8608.

To conclude, the proposed conic approach managed to solve all types of problems presented in Example 1.1, including the NCM problem (3), the rank-constrained NCM problem (4) and the correlation matrix completion. In the following, we want to demonstrate the performance of the proposed conic approach in solving the general NCM problem (3) and the rank-constrained NCM problem (4).

4.4.2 Solving the NCM problem

In this set of experiments, we compared the proposed SDP reformulation (78) of the NCM problem (3) with the alternating projections algorithm and the preconditioned Newton method implemented in the NAG library in Matlab. We generated 100 random empirical correlation matrices C of order $n = 10$, $n = 20$, and $n = 50$. The results of these experiments, shown in Table 5, demonstrate that for a larger n, the alternating projections algorithm does not converge to a solution in the first 200 iterations. Therefore, it is more effective to use the Newton preconditioned method, as suggested in [16].

Although our SDP approach requires more computational time than the two standard approaches due to the need to initialize the CVX modeling system, it still managed to find an optimal solution with the same (or even slightly lower) optimal value than the other two methods for the NCM problem (3). However, for practical purposes, it is more efficient to use the preconditioned Newton method. Moreover, in all 100 experiments, the deviation between the approximations obtained by the SDP approach and the preconditioned Newton method was at the level of 10^{-7}, indicating the accuracy and reliability of the SDP approach.

n	time (AP)	time (N)	time (SDP)	$\|X_{ap} - C\|_F$	$\|X_n - C\|_F$	$\|X_{sdp} - C\|_F$
10	0.005 s	0.005 s	0.107 s	2.585818121	2.585818303	2.585818104
20	0.008 s	0.005 s	0.172 s	6.634680585	6.634680745	6.634680550
50	x	0.005 s	0.107 s	x	20.3988005	20.398780

Table 5: Comparison of the existing methods and the SDP reformulation in solving NCM problems. Comparison of solution methods for the NCM problem (3): the alternating projections algorithm (AP), the preconditioned Newton method (N), and SDP reformulation (SDP). The table displays the average computation time and average optimal values obtained over 100 randomly generated correlation matrices of order $n = 10, 20, 50$. The results are taken from our paper [44].

4.4.3 Solving the rank-constrained NCM problem

In this part, we address the rank-constrained NCM problems of the form (4). First, we investigate the performance of the bisection algorithm (Algorithm 4) using two different methods to solve the modified rank-constrained problem (68) in each iteration: the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3). To test these methods, we generate empirical correlation matrices $C \in \mathbb{S}^n$ of various sizes n and set different target rank k.

Tables 6 and 7 present the results of applying two different methods, namely the convex iteration and the log-det heuristic, to solve the rank-constrained NCM problem (4) via the bisection algorithm (Algorithm 4). The goal is to find the best possible approximation of ε-rank k for various values of k and three different empirical correlation matrices of sizes $n = 20$, $n = 30$, and $n = 40$.

The bisection algorithm employs the SDP relaxation (3) and the bi-criterion versions of the convex iteration (82), (83), and the bi-criterion of the log-det heuristic (81). As shown in Table 6 and Table 7, the computation time, rank, and objective value $g(X)$ increase with the size n of the empirical correlation matrix C. Unexpectedly, the log-det heuristic fails to converge to a solution with ε-rank equal to k when the difference between n and k is significant. Therefore, the convex iteration version of the bisection algorithm

n	k	semidefinite relaxation			convex iteration			bisection algorithm		
		time (s)	rank	g_0	time (s)	iter,	g_1	time (s)	iter.	$\hat{g}$
20	5	0.17	9	41.87	0.63	1	48.68	223.82	24	48.51
	4				1.24	2	60.74	312.81	26	60.04
	3				1.23	2	84.24	282.98	27	82.93
	2				1.36	2	146.81	327.20	28	141.88
30	5	0.44	13	122.09	2.13	2	172.30	667.10	27	169.10
	4				2.18	2	215.92	669.44	28	207.97
	3				2.13	2	292.84	711.10	29	277.47
	2				2.25	2	455.26	816.29	30	427.74
40	5	1.14	16	248.35	4.36	2	353.53	1496.8	28	350.41
	4				4.32	2	429.49	1545.8	29	422.82
	3				4.50	2	568.94	1604.6	30	559.47
	2				6.68	3	822.99	1809.6	31	812.28

Table 6: Solving rank-constrained NCM problems of different sizes using the convex iteration. Comparison of the results of the SDP relaxation (78), the convex iteration (Algorithm 3), and the bisection algorithm (Algorithm 4) for solving the rank-constrained NCM problem (4). The bisection algorithm used the convex iteration to solve the modified rank-constrained problem in each iteration and started from the initial interval $[g_0, g_1]$. Both the convex iteration and the bisection algorithm found solutions with the ε-rank equal to k.

(Algorithm 4) can be considered a more reliable method. Moreover, it starts with a tighter interval, which can save one iteration. However, this method is more time-consuming as it solves two optimization problems at each iteration. As in Subsection 4.4.1, we note that the longer computation time taken by the bisection algorithm (Algorithm 4) is associated with the choice of M for the stopping criteria.

4.4.4 Comparison of methods for solving the rank-constrained feasibility problems

We aim to investigate whether the observations from Figure 8 can be generalized to other scenarios. To achieve this, we generate 100 empirical correlation matrices of size

n	k	semidefinite relaxation			log-det heuristic			bisection algorithm		
		time (s)	rank	g_0	time (s)	iter.	g_1	time (s)	iter.	$\hat{g}$
20	5	0.17	9	41.88	0.78	2	51.60	197.71	25	48.51
	4				2.91	8	67.95	211.82	26	60.05
	3				x	x	x	x	x	x
	2				x	x	x	x	x	x
30	5	0.44	13	122.09	3.90	6	188.25	556.57	27	170.19
	4				x	x	x	x	x	x
	3				x	x	x	x	x	x
	2				x	x	x	x	x	x
40	5	1.14	16	248.35	x	x	x	x	x	x
	4				x	x	x	x	x	x
	3				x	x	x	x	x	x
	2				x	x	x	x	x	x

Table 7: Solving rank-constrained NCM problems of different sizes using the log-det heuristic. Results of the SDP relaxation (78), the log-det heuristic (Algorithm 1) and the bisection algorithm (Algorithm 4) using the log-det heuristic for solving modified rank-constrained NCM problems in each iteration of Algorithm 4. The cases, when the log-det heuristic failed to find a solution with ε-rank equal to k, are labeled by "x". In such cases, the bisection algorithm was not used since missing an input.

$n = 20$ and solve the corresponding rank-constrained NCM problems of the form (80) for $k = 4$. We chose $n = 20$ and $k = 4$ to ensure that our bisection algorithm (Algorithm 4) works for both its versions. The averaged results of these experiments are presented in Table 8. Since the SDP relaxation (3) failed to find a rank-4 solution in any of the 100 generated problems, the usage of rank minimization heuristics and rank reduction algorithms is reasonable. Interestingly, the objective value $g(X)$ is significantly larger than $\|C - X\|_F^2$ in the case of the original log-det heuristic (Algorithm 1) and the original convex iteration (Algorithm 3). Although they offer a wider interval $[g_0, g_1]$ for Algorithm 4, we could make it significantly tighter by using $[g_0, \|C - X\|_F^2]$. However, the bi-criterion versions of these methods still provide a better value of $\|C - X\|_F$ and therefore a better

method	$rank(X)$	$g(X)$	$\|X - C\|_F$	time (s)	%	empirical ε	iters
SDP relaxation	9.94	44.45	6.66	0.31	0	2.09	1
rank reduction	4	113.72	10.66	1.13	100	9.32e-18	5.96
log-det heuristic	3.94	3177.18	9.52	0.92	100	1.11e-07	1.92
log-det ($\alpha = 100$)	4	81.26	9.00	1.40	100	4.10e-08	3.19
convex iteration	4	4254.13	9.10	1.18	100	4.30e-10	1
convex iter. ($\alpha = 100$)	4	73.38	8.55	1.17	100	3.00e-10	1

Table 8: Solving rank-constrained NCM problems using various methods. Comparison of methods for solving 100 rank-constrained NCM problems with $k = 4$ and $C \in \mathbb{S}^{20}$, including the SDP relaxation (3), the rank reduction algorithm (Algorithm 4), the (bi-criterion) log-det heuristic (Algorithm 1) and the (bi-criterion) convex iteration (Algorithm 3). The table displays average performance metrics, including the success rate in finding a rank-k solution denoted by "%".

rank-4 approximation of the given empirical correlation matrix C. Table 8 validates our hypothesis from Figure 8 that the rank reduction algorithm (Algorithm 2) provides the worst rank-4 approximation of C. In Section 4.4.6, we explore its performance when addressing larger problems.

The results presented in Table 9 are remarkable as they show that the convex iteration (Algorithm 3) achieved a solution of ε-rank equal to $k = 4$ after only one iteration. Despite a lower number of iterations, the computation time is comparable with the log-det heuristics. These findings validate the comparison in Figure 8, as the best performance was obtained by the bi-criterion version of the convex iteration (Algorithm 3 with (66)).

4.4.5 Bisection algorithm performance

Let us investigate the performance of the bisection algorithm (Algorithm 4) using the inputs from Table 8 and summarize the results in Table 9. As expected, when the initial interval is wide, such as in the case of using original versions of methods, the relative improvement provided by $\hat{g}$ is low (see the first column of Table 9). The number of iterations required depends on the length of the initial interval $[g_0, g_1]$. However, the computation time is most affected by those iterations of the bisection algorithm that do not provide a rank-k solution, as this is only recognized after M "constant-rank" iterations. In the last columns of Table 9, we report the number of generated problems (out of 100) in which each method found a rank-4 solution providing the best value of these criteria among the considered methods. Note that in many cases, both versions of the log-det heuristic, as well as both versions of the convex iteration, found the same values. When comparing only the bi-criterion versions of the log-det heuristic and the convex iteration, the bi-criterion log-det heuristic provides a better value $\hat{f}$ in only 4 cases, with an average deviation of 0.016. On the other hand, the bi-criterion convex iteration is better in 96 cases, with an average deviation of 0.047. These experiments suggest that the convex iteration is a better tool to solve modified rank-constrained problems (68) in Algorithm 4.

4.4.6 Choice of relative weights

In the concluding section, we present the results published in [44], where we investigated the rank reduction algorithm (Algorithm 2) to find a feasible solution and the bi-criterion version of the convex iteration (Algorithm 3 with (66), (56)). Instead of a constant relative weight α, we use different values of $\alpha(t)$ in each iteration t. However, solving a problem with a large size n and a very low desired rank k may cause the convergence of these algorithms to require many iterations. This could be problematic since the convex iteration solves two optimization problems in each iteration.

For the next experiment, we generate a random correlation matrix of size $n = 100$, and consider a three-factor model, which involves solving the rank-constrained NCM problem (4) for $k = 3$. In the first step, we solve the rank-constrained NCM problem (80) using the convex iteration (82) and (83). Since one of the problems is a bi-criterion problem, selecting a suitable relative weight $\alpha(t)$ is essential. We set $\alpha(t)$ as an increasing

method	$\lvert 1-\frac{g_1-\hat{g}}{g_1-g_0}\rvert$	$\lvert 1-\frac{f_1-\hat{f}}{f_1-f_0}\rvert$	iters	time (s)	# of min($\hat{g}$)	# of min($\hat{f}$)
log-det heuristic	0.0159	0.4336	32.93	210.92	4	4
log-det ($\alpha = 100$)	0.6815	0.7149	26.69	212.79	3	4
convex iteration	0.0058	0.6694	33.22	309.35	79	87
convex iter. ($\alpha = 100$)	0.8413	0.8344	26.18	305.98	78	87

Table 9: Solving rank-constrained NCM problems using the bisection algorithm. Comparison of the performance of the bisection algorithm (Algorithm 4) when using different sources of inputs and methods to solve modified rank-constrained problems in each iteration. The results are based on 100 generated rank-constrained NCM problems of the form (4), and the table displays the average values computed. The first two columns correspond to the relative improvement of $g_i = tr(Z_i)$ and $f_i = f(X_i) = \|C - X_i\|_F$.

sequence of the number of iterations t and compare its behavior. Figure 10 and Table 10 indicate that $\alpha(t) = t$ requires 72 iterations to find a rank-3 solution, making it slow, as two optimization problems are solved in each iteration. Using a faster-increasing sequence results in fewer iterations for the convex iteration, while still achieving a comparable value of the objective function $\|C - X\|_F$.

Subsequently, we apply the rank reduction algorithm to solve the rank-constrained NCM problem (80). As shown in Figure 10 and Table 10, the rank reduction algorithm provides a much better rank-3 approximation of the empirical correlation matrix C than the convex iteration (82),(83).

Remarkably, the rank reduction algorithm (Algorithm 2) discovered a solution of rank k, despite the fact that it is only guaranteed to provide solutions of rank lower or equal to $\lfloor \frac{\sqrt{8n+1}-1}{2} \rfloor = 13$. These results imply that the rank reduction algorithm (Algorithm 2) may be a valuable tool when working with large n.

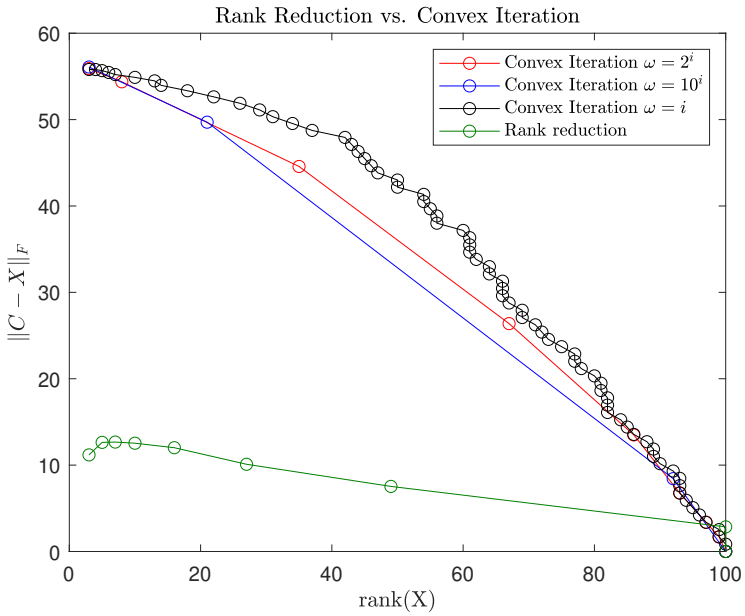


Figure 10: Trade-off graph between the rank and the objective function. Trade-off graph between the rank and the objective function value of the solution found by the rank reduction algorithm (green) and the convex iteration for relative weights $\alpha(t) = 2^t$ (red), $\alpha(t) = 10^t$ (blue) and $\alpha(t) = t$ (black) with t being the number of iteration, while $n = 100$ and $k = 3$. Points represent solutions in particular iterations. The starting point $[100, 0]^T$ represents the given full-rank correlation matrix C. The figure is taken from our paper [44] where $\alpha(t)$ is labeled as $\omega(i)$.

	Convex Iter. $\alpha(t) = 2^t$	Convex Iter. $\alpha(t) = 10^t$	Convex Iter. $\alpha(t) = t$	Rank Reduction
# iterations	8	3	72	8
$\|X - C\|_F$	55.90	56.09	55.80	11.18

Table 10: Comparison of the rank reduction algorithm and the convex iteration regarding various choices of relative weights. Comparison of the rank reduction algorithm and the convex iteration for $\alpha(t) = 2^t$, $\alpha(t) = 10^t$ and $\alpha(t) = t$, where t is the number of iterations, $n = 100$ and $k = 3$.

5 Procrustes problems

The Procrustes problems (PPs) are a well-known class of optimization problems, named after a character in Greek mythology called Procrustes [54]. Procrustes was a bandit who would stretch or cut off the limbs of his victims to make them fit his iron bed. Similarly, in Procrustes problems, the goal is to stretch or compress one set of data to make it fit another set of data. Fortunately, solving Procrustes problems is not as painful as Procrustes' methods.

Recall the generalized formulation of the matrix approximation problem

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & f(X) := \|W \circ (C - AXB)\| \\ & X \in \mathcal{P}, \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{p \times m}, B \in \mathbb{R}^{n \times q}, W, C \in \mathbb{R}^{p \times q}$ are the data and $X \in \mathbb{R}^{m \times n}$ is the matrix variable. A general Procrustes problem is formulated as the generalized matrix approximation problem (1), where at least one of the given matrices $A \in \mathbb{R}^{p \times m}$ and $B \in \mathbb{R}^{n \times q}$ does not equal identity. It is worth noting that in standard matrix approximation problems $A = I_m$ and $B = I_n$, such as in the problem of finding the nearest correlation matrix (3) from Chapter 4.

In standard Procrustes problems, the matrix B is assumed to be equal to the identity matrix I_n, implying that the residuals $C - AX$ have equal variance. However, in this chapter, we also consider a more general case of weighted Procrustes problems, as introduced in publications such as [97, 65, 74]. The classification of Procrustes problems as *balanced* or *unbalanced* can be made based on the dimension of the matrix variable, as described in previous works like [83, 104, 105]. Balanced Procrustes problems refer to cases where $m = n$, meaning that the matrix variable is squared, while unbalanced Procrustes problems refer to cases where $m \neq n$, indicating that the matrix variable is rectangular. In this chapter, we adopt this terminology.

As discussed in Chapter 2, the choice of matrix norm in the objective function depends on the specific problem being solved and the characteristics of the data being analyzed. In the objective of the Procrustes problem (1), the Frobenius norm is commonly used [87, 104]. Some authors, such as [97] and [98], consider the l_1 norm, which is a robust alternative to the least squares. However, our conic approach also covers the l_∞ norm,

and the spectral norm l_2, as introduced in Chapter 2.

A typical Procrustes problem is formulated over orthogonal matrices, that is, the feasible set $\mathcal{P}$ is a matrix manifold. Some authors also consider other types of feasible sets, such as the set of positive semidefinite matrices, such as [6]. Similar to the generalized matrix approximation problem (1), we allow for linear, semidefinite, quadratic, and rank constraints to define the feasibility set $\mathcal{P}$ (see Assumption 1.1). This means that we cover all the aforementioned classes as well as other challenging cases that are difficult to handle using standard approaches. These may include orthogonal or oblique Procrustes problems with additional linear constraints or Procrustes problems over the set of projection matrices.

In the following sections, we discuss the most common types of Procrustes problems, including orthogonal, oblique, semidefinite, and projection PPs. For each subclass, we provide an overview of the existing solution approaches. These approaches are summarized in Table 11, which highlights the diversity of solution algorithms that have been proposed for Procrustes problems, depending on the specific matrix norm and constraints involved, as well as the robustness of our proposed conic approach. Furthermore, we propose the (rank-constrained) SDP reformulations for these subclasses of Procrustes problems and demonstrate their correctness by numerical experiments.

The computations were carried out using MATLAB R2019a [71] on a laptop equipped with the 11th Gen Intel(R) Core(TM) i7-1165G7 processor running at 2.80GHz. To solve the SDP programs, we utilized the SDPT3 solver, which is included in CVX, a package for specifying and solving convex programs [56, 55]. The rank of a matrix was determined as the ε-rank according to Definition 3.1 with respect to $\varepsilon = 10^{-6}$. For some experiments, we determine an "empirical" ε as the k-th largest eigenvalue of a solution, that is, if we set ε equal to this value, the particular algorithm would find a solution with ε-rank equal to k. This value can be useful to analyze the performance of the particular algorithm or the quality of a found solution. It is important to note that higher accuracy would not change the dynamics of the solution method, but it may require more computation time[5].

[5]Our Matlab codes with implemented methods are available at `https://github.com/TereziaF/A-conic-approach-for-solving-matrix-approximation-problems`

class	type	norm	solution method	source	conic
OPP	balanced	Frob	explicit solution via SVD	[87]	✓
			eigenvalue decomposition	[83]	
			SDP relaxation	[1]	
	unbalanced	Frob	relaxation-based methods	[15]	✓
			Newton-type methods	[99], [33]	
			successive projection	[105]	
			SVD-based OLSR method	[107]	
			eigenvalue-based method	[104]	
			SDP relaxation	[28]	
	weighted	Frob	extension of standard methods	[99], [41], [42]	✓
			differential approach	[24], [23]	
		l_1	differential approach	[97]	✓
		l_2, l_∞			✓
ObPP	standard	Frob	projection method	[54]	✓
			differential approach	[96]	
		l_1	separation of problem	[98], [14]	✓
	weighted	l_1	differential approach	[98]	✓
		Frob, l_2, l_∞			✓
SDPP		Frob	necessary and sufficient conditions	[6], [50], [62]	✓
		l_1, l_∞, l_2			✓
proj. PP					✓
add.cons.					✓

Table 11: Solution methods for different types of Procrustes problems. Solution methods for orthogonal (OPP), oblique (ObPP), semidefinite (SDPP) and projection (proj.PP) Procrustes problems. The last column indicates classes covered by the proposed conic approach. The shortcut "add.cons." means an arbitrary class of PPs with additional linear or semidefinite constraints.

5.1 Orthogonal Procrustes problems

One of the most well-known subclasses of Procrustes problems is the class of *orthogonal Procrustes problems* (OPPs), where the matrix variable is assumed to be orthogonal, or at least having orthogonal columns (rows). OPPs are used to find an orthogonal matrix that maps one set of data to fit another set of data, by means of a specific matrix norm. OPPs have applications in various areas such as rigid body dynamics [12, 88], psychometrics [87, 99], multidimensional scaling [25], or global positioning system [10]. Moreover, unbalanced OPPs map high dimensional data (with dimension m) into a space with a lower dimension $n << m$. This applies, e.g. in the orthogonal least square regression, which may be used for feature extraction [104, 107]. The weighted OPPs, which involve a general matrix B, find applications in multivariate analysis [52].

A general orthogonal Procrustes problem is formulated as follows

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & f(X) := \|W \circ (C - AXB)\| \\ & X^T X = I_n, \end{aligned} \tag{85}$$

where $A \in \mathbb{R}^{p \times m}, B \in \mathbb{R}^{n \times q}, W \in \mathbb{R}^{p \times q}$ and $C \in \mathbb{R}^{p \times q}$ are the given data. Note that the matrix variable $X \in \mathbb{R}^{m \times n}$ is constrained to have orthogonal columns.

It is worth noting that the nonconvex quadratic constraint in (85) is commonly relaxed by imposing the condition $X^T X \succeq I_n$. However, its equivalent reformulation is obtained using Lemma 2.1, which states that

$$X^T X = I_n \;\; \Leftrightarrow \;\; V = \begin{pmatrix} I_m & X \\ X^T & I_n \end{pmatrix} \succeq 0 \;\wedge\; rank(V) = m. \tag{86}$$

***Remark* 1.** *Assuming standard OPPs with the Frobenius norm, the objective* $\|C - AX\|_F$ *can be rewritten as follows*

$$\|C - AX\|_F = tr\Big[(C - AX)(C - AX)^T\Big] = tr(CC^T) + tr(AXX^TA^T) - 2tr(C^TAX). \tag{87}$$

Since the first term is constant, the objective can be reduced to

$$tr(AXX^TA^T) - 2tr(C^TAX). \tag{88}$$

This feature is used in most of the existing solution methods, some of which are described in the following subsection. Note that if $m = n$*, we get a linear function that enables the derivation of an explicit solution.*

5.1.1 Known approaches for solving OPPs

The standard balanced OPP, where $m = n$, $B = I_n$ and the goal is to find an orthogonal matrix X that minimizes $\|C - AX\|_F$, was studied in [87]. It was shown that this problem has a closed-form solution, which can be obtained by the singular value decomposition. Specifically, the optimal solution X^* is given by

$$X^* = VU^T \tag{89}$$

where U and V are orthogonal matrices obtained from the singular value decomposition of $C^T A$. Subsequent publications have focused on accelerating the computation of the solution. For example, in [83], a method based on eigenvalue decomposition is proposed to speed up the computation of the solution of the standard balanced OPP.

Unlike standard balanced OPPs with the Frobenius norm in the objective, any other subclass of PPs does not have a known closed-form solution. Therefore, it is necessary to use specific minimization algorithms. In the following, we discuss the existing solution methods for solving different subclasses of OPPs, which are also summarized in Table 11.

Several algorithms have been proposed to solve unbalanced OPPs on the Stiefel manifold. One approach is to use relaxation-based iterations, as proposed in [15]. This method involves relaxing the orthogonality constraint on X and solving a sequence of relaxed subproblems iteratively until convergence is achieved. Another approach is to use Newton-type iterations, as proposed in [33, 99], which involve using the Newton method on manifold to update X in each iteration until a local optimum is reached. In addition to iterative methods, necessary and sufficient conditions for local optimality in unbalanced OPPs have been derived in [34]. These conditions provide insights into the properties of optimal solutions and can be used to guide the development of optimization algorithms for solving unbalanced OPPs.

The special case of unbalanced OPPs with $n = 1$ is known as the trust-region subproblem of the trust-region method in optimization [75]. This knowledge was used in [105] to design the successive projection method, where all but one column of X are fixed, and a trust-region subproblem is solved in each iteration. In [107], an iterative OLSR algorithm, based on the use of singular value decompositions was introduced for solving the orthogonal least square regression, which has shown effectiveness in practice. More

recently, in 2020, an eigenvalue-based approach was introduced in [104] that outperforms the successive projection method from [105]. Specifically, the authors proposed an iterative algorithm based on the self-consistent-field (SCF) iteration, which is an efficient method for solving eigenvector-dependent nonlinear eigenvalue problems.

Weighted OPPs are another interesting case, where A and B are general matrices and the goal is to minimize $\|C - AXB\|_F$. Several algorithms have been developed for solving these problems based on the extension of standard algorithms to the case of Stiefel manifolds [99, 41, 42] and were demonstrated to be effective in computation. In addition, in [24] and [23], an approach based on solving differential equations has been introduced for weighted OPPs with the Frobenius norm in the objective. In [97], this approach was extended to solve also weighted OPPs with the l_1 norm in the objective. However, the numerical experiments were executed only for a few small examples and the computation time was not specified.

Previous attempts to solve OPPs using a conic optimization approach have been reported in the literature, but they have been limited to standard OPPs with the Frobenius norm in the objective. In [1], a relaxation-based approach was proposed for standard balanced OPPs with potential data uncertainties. Another SDP relaxation was proposed in [28] to handle standard unbalanced OPPs. In this work, the authors exploited the observation of Remark 1 and utilized vectorization to obtain the following SDP relaxation for standard OPPs:

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}, Y\in\mathbb{S}^m} \quad & tr(A^TAY) - 2tr(C^TAX) \\ & \begin{pmatrix} I_n & X^T \\ X & Y \end{pmatrix} \succeq 0, \\ & \begin{pmatrix} I_m & X \\ X^T & I_n \end{pmatrix} \succeq 0, \\ & tr(Y) = n. \end{aligned} \tag{90}$$

In contrast to the existing methods, our approach has several advantages. First, it can handle both standard and weighted OPPs, which was not possible with earlier methods. Moreover, it allows for various matrix norms in the objective, such as the l_1 norm and the spectral norm, making it useful for applications where outliers are present, such as orthogonal least squares regression, as noted in [107]. In addition, our method can handle

additional linear and semidefinite constraints that may arise in problem formulation (1), which is not possible with existing methods. However, it is important to note that our approach may require more computation time compared to existing methods. This is because our approach is designed to handle a wider class of PPs, rather than a specific subclass and relies on solving conic problems. Despite this, the main advantage of our approach is its ability to handle challenging subclasses of PPs, which we discuss in the upcoming sections.

5.1.2 The proposed conic approach

In this subsection, we present reformulations of the general OPP (85) with respect to different types of matrix norms in the objective.

After applying the statement of Lemma 2.1 to the general OPP (85), we rewrite the nonconvex quadratic constraint on the orthogonality of the matrix variable as in (86) to obtain this reformulation:

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}, V\in\mathbb{S}^{m+n}} \quad & f(X) := \|W \circ (C - AXB)\| \\ & V = \begin{pmatrix} I_m & X \\ X^T & I_n \end{pmatrix} \succeq 0, \\ & rank(V) = m. \end{aligned} \tag{91}$$

If the objective of (91) is defined in terms of the Frobenius norm, we apply Proposition 2.5 to have

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}, V\in\mathbb{S}^{m+n}, Z\in\mathbb{S}^p} \quad & tr(Z) \\ & V = \begin{pmatrix} I_m & X \\ X^T & I_n \end{pmatrix} \succeq 0, \\ & rank(V) = m, \\ & \begin{pmatrix} I_q & (W \circ (C - AXB))^T \\ W \circ (C - AXB) & Z \end{pmatrix} \succeq 0. \end{aligned} \tag{92}$$

In the case of the l_1 norm in the objective of (91), using Proposition 2.2 we get

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n},V\in\mathbb{S}^{m+n},t\in\mathbb{R},S\in\mathbb{R}^{p\times q}} \quad & t \\ V = \begin{pmatrix} I_m & X \\ X^T & I_n \end{pmatrix} &\succeq 0, \\ rank(V) &= m, \\ -S \leq W \circ (C - AXB) &\leq S, \\ S^T\mathbf{1}_p &\leq t\mathbf{1}_q. \end{aligned} \tag{93}$$

If the objective of (91) contains the l_∞ norm, we apply Proposition 2.3 to obtain

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n},V\in\mathbb{S}^{m+n},t\in\mathbb{R},S\in\mathbb{R}^{p\times q}} \quad & t \\ V = \begin{pmatrix} I_m & X \\ X^T & I_n \end{pmatrix} &\succeq 0, \\ rank(V) &= m, \\ -S \leq W \circ (C - AXB) &\leq S, \\ S\mathbf{1}_q &\leq t\mathbf{1}_p. \end{aligned} \tag{94}$$

If (91) involves the spectral norm, Proposition 2.4 provides the following reformulation

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n},V\in\mathbb{S}^{m+n},s\in\mathbb{R}} \quad & s \\ V = \begin{pmatrix} I_m & X \\ X^T & I_n \end{pmatrix} &\succeq 0, \\ rank(V) &= m, \\ \begin{pmatrix} sI_p & W \circ (C - AXB) \\ (W \circ (C - AXB))^T & sI_q \end{pmatrix} &\succeq 0. \end{aligned} \tag{95}$$

Note that regardless of the type of matrix norm being minimized, the general OPP (85) can be reformulated as a rank-constrained SDP problem, which can be solved using algorithms presented in Chapter 3. However, due to the unit diagonal of the block matrix $V \in \mathbb{S}^{m+n}$, rank of which is being constrained, the trace heuristic (39) is useless since it is equivalent to the SDP relaxation (34). In addition, reformulating these rank-constrained SDP problems into the standard form is not straightforward, making the rank reduction algorithm (Algorithm 2) nontrivial to use. Hence, in the numerical part, we solve these problems only using the SDP relaxation (34), the log-det heuristic (Algorithm 1), the convex iteration (Algorithm 3), and the bisection algorithm (Algorithm 4) in their various versions.

5.1.3 Numerical results

This section presents an overview of the numerical results obtained by solving different types of OPPs. We first solve standard balanced and unbalanced OPPs with the Frobenius norm in the objective, for which either an explicit solution or an effective solution method is known, to demonstrate the correctness of the proposed conic approach. Next, we apply the conic approach to solve weighted OPPs with various matrix norms in the objective, including not only the Frobenius norm but also the l_1 norm, l_∞ norm, and the spectral norm. Finally, we address OPPs with additional linear constraints.

In the experiments, we use two types of OPPs generation. In both, we first generate an orthogonal matrix using the built-in function `RandOrthMat` from Matlab libraries. The first type of generated problems has a zero optimal value. We randomly generate A and B, and determine C such that the generated X solves the equation system $C = AXB$. For this type of problem, we know the optimal value a priori, enabling us to evaluate the accuracy of the solution. The second type of generated problems has an optimal value slightly deviated from zero. We achieve this by generating a matrix $\Delta \in \mathbb{R}^{p\times q}$ from $N(0,1)$ and setting $C = AXB + 0.5\Delta$, as suggested in [24]. We distinguish between these two types of generated problems in the tables by denoting $f^* = 0$ and $f^* \neq 0$, respectively. To assess the feasibility of the obtained solution X, we utilize the criterion $\|X^TX - I_n\|_F$. In all experiments, we apply the algorithms presented in Chapter 2 with the following inputs: $\varepsilon = 10^{-6}$, $\delta = 0.01$, $\rho = 10^{-6}$, and $M = 10$, unless specified otherwise.

5.1.3.1 Application - Evaluating the accuracy of an ancient map

Recall the assignment of Example 1.2. The task is to find an orthogonal matrix $X \in \mathbb{R}^{2\times 2}$, a scaling factor $\rho \in \mathbb{R}$ and a translation vector $d \in \mathbb{R}^2$ to evaluate the accuracy of an ancient map with respect to a modern map by solving a balanced OPP of the form (5).

We searched for the orthogonal transformation matrix X using both the explicit solution based on the singular value decomposition (89), and our rank-constrained SDP reformulation (92). In this case, the SDP relaxation of (92) was sufficient to find an orthogonal solution

$$X^* = \begin{pmatrix} 0.983 & -0.182 \\ 0.182 & 0.983 \end{pmatrix}.$$

method	$\|C - AX\|_F$	$\|XX^T - I\|_F$	$rank(Y)$	time
SVD	3867.2	7.83×10^{-16}	2	0.01
SDP relaxation	3867.2	8.64×10^{-8}	2	0.22

Table 12: Results for application – evaluating the accuracy of an ancient map. Comparison of the explicit solution based on the SVD (89) and the SDP relaxation of (92) in solving Example 1.2.

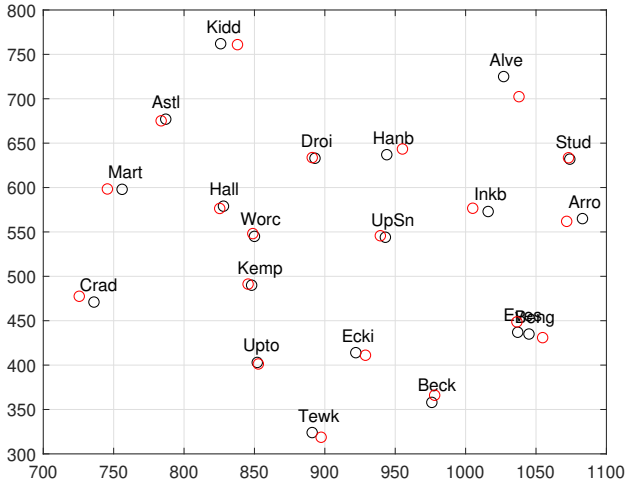


Figure 11: Transformed locations obtained as a result of solving Example 1.2. Transformed ancient data (red) fitting modern data (black) from Example 1.2.

A comparison of these two approaches is summarized in Table 12. Despite the longer computation time, the results obtained by the SDP relaxation correspond to the explicit solution. We can easily verify that $det(X^*) = 1$ indicating that X^* is a rotation matrix of the form

$$X^* = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},$$

where $\theta = 10.2°$. This angle represents the difference in orientation between the ancient map and the modern map. Consequently, the transformation matrix X^* is used to determine $\rho^* = 2.36$ and $d^* = (502.67, 296.03)^T$ using (7). The optimal value of (5), that is, the value of $\|C - (\mathbf{1}_2 d^T + \rho AX)\|_F$, is 44.56, which denotes the sum of all absolute deviations between locations. Comparing this value with the value of $\|C - AX\|_F$ in Table 12, we can conclude that translation and scaling significantly improved the fit of the data.

	explicit solution	semidefinite relaxation	desired value
$\|C - AX\|_F$	3.222e-13	2.9427e-05	0
$\|X^T X - I_n\|_F$	1.8128e-15	1.4927e-06	0
$rank(V)$	5	5	5
time (sec)	0.00003	0.43271	

Table 13: Solving standard balanced OPPs with the Frobenius norm in the objective. Results for 100 generated standard balanced OPPs with the Frobenius norm in the objective, including average values of the criteria for optimality and orthogonality, rank of the block matrix V, computation time, and desired value for each criterion.

In Figure 11, black-dot locations from the modern map are fitted with our transformed red-dot ancient data. These results confirm a quite high accuracy of the ancient map and the correctness of the proposed reformulation (92).

5.1.3.2 Standard balanced OPPs with the Frobenius norm and the spectral norm in the objective

Although in the previous application, the solution found by the SDP relaxation (92) corresponded to the explicit optimal solution (89), we aim to compare these methods also for 100 generated standard balanced OPPs of different sizes with the optimal value equal to zero. After that, we will observe changes in optimal values, values of the orthogonality criterion, and computation time obtained for 100 generated standard balanced OPPs with the spectral norm in the objective.

In Table 13, we provide a summary of the results obtained by comparing the SDP relaxation of (92) with the explicit solution based on the singular value decomposition (89). The first row of the table confirms that the average reached optimal value equals zero within the specified tolerance, indicating the ε-optimality of the solutions. The second row demonstrates that the found solutions are orthogonal, providing $rank(V)$ equal to the desired value of $m = 5$.

$p\backslash n$	5	10	15	20
25	2.6251e-09	1.2566e-10	1.3090e-09	7.5200e-10
50	2.4351e-09	2.0353e-10	3.9213e-10	1.0291e-09
75	1.9186e-09	1.6807e-10	2.9516e-10	1.7583e-10
100	1.9322e-09	1.5407e-10	2.7570e-10	1.9705e-09

Table 14: Accuracy of the optimal values of balanced OPPs with the spectral norm in the objective. Average values of the objective function $\|C - AX\|_2$ obtained by the SDP relaxation of (95) applied to 100 generated balanced OPPs with the spectral norm in the objective for different values of p and n.

$p\backslash n$	5	10	15	20
25	2.4643e-10	2.1996e-11	3.5808e-10	3.8831e-10
50	1.5312e-10	2.0211e-11	5.3254e-11	1.8592e-10
75	9.7703e-11	1.3033e-11	2.9904e-11	2.2719e-11
100	8.6725e-11	1.0144e-11	2.3278e-11	2.1241e-10

Table 15: Accuracy of orthogonal solutions of balanced OPPs with the spectral norm in the objective. Average values of the orthogonality criterion $\|X^T X - I_n\|_F$ obtained by the SDP relaxation of (95) applied to 100 generated balanced OPPs with the spectral norm in the objective for different values of p and n.

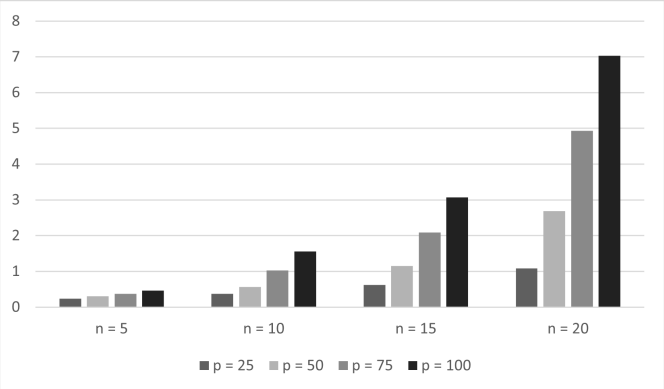


Figure 12: Average computation time for balanced OPPs with the spectral norm. Average computation time (in seconds) of the SDP relaxation applied to solve 100 generated balanced OPPs with the spectral norm in the objective for different values of p and n.

Although the computation time for finding the explicit solution is naturally lower compared to the SDP relaxation of (92), it is irrelevant for our purposes, as we are only evaluating the correctness of the proposed conic approach. The experiments were performed with a specified tolerance $\varepsilon = 10^{-6}$. However, if a more accurate solution is required, the SDP relaxation can be improved using modified methods and rank reduction algorithms presented in Section 3.2.

We conducted numerical experiments to investigate the impact of changing the number of rows of $C \in \mathbb{R}^{p\times n}$ and $X \in \mathbb{R}^{n\times n}$ in standard balanced OPPs with the spectral norm in the objective. For each combination of p and n, we generated 100 problems, and the results are presented in Table 14 and Table 15. The tables demonstrate that the SDP relaxation of (95) provided an optimal solution for all generated problems of this type, with a chosen tolerance $\varepsilon = 10^{-6}$. Additionally, we analyzed the behavior of computation time with respect to the parameters p and n, as shown in Figure 12.

5.1.3.3 Application - Feature extraction

Consider the assignment of Example 1.3, where we have provided a detailed description of the Yalefaces data set, consisting of $p = 165$ images with $m = 256$ features corresponding to $n = 15$ individuals. The objective is to extract features from this data set using the orthogonal least squares regression, that is, solving an unbalanced OPP of the form

$$\begin{aligned} \min_{X\in\mathbb{R}^{256\times 15}} \quad & \|C - AX\|_F \\ & X^T X = I_n, \end{aligned} \tag{9}$$

where $C \in \mathbb{R}^{165\times 15}$ and $A \in \mathbb{R}^{165\times 256}$.

To solve this problem, we used a method introduced in [107], called OLSR which involves iteratively performing singular value decompositions[6]. Our aim is to assess the accuracy of the proposed rank-constrained SDP reformulation (92) by comparing our results with those obtained by the OLSR algorithm.

Table 16 presents a comparison between the results obtained by the OLSR algorithm [107] and the proposed conic approach for solving (9) applied to extract features from the Yalefaces data set. Initially, we applied the SDP relaxation of (92) to solve (9). However,

[6]Scripted algorithm accessible on the website `https://github.com/StevenWangNPU/OLSR_NC2016`.

norm	criterion	OLSR	SDP relaxation	modified log-det heuristic
Frobenius	$\|C - AX\|_F$	2.9571	2.9570	2.9570
	$\|X^T X - I_n\|_F$	4.3586e-15	1.0000	1.1725e-08
	$rank(V)$	256	258	256
	time (s)	0.4719	817.6205	1534.89
l_1	$\|C - AX\|_1$	x	7.2534	7.2534
	$\|X^T X - I_n\|_F$	x	0.3560	1.6922e-9
	$rank(V)$	x	270	256
	time (s)	x	51.58	106.2779

Table 16: Results for application – feature extraction. Results of the unbalanced OPP (10) applied to solve the orthogonal least squares regression using the Yalefaces data set, introduced in Example 1.3. Comparison of the OLSR algorithm [107], the SDP relaxation of (92), and the modified log-det heuristic (Algorithm 1 with (70)) applied to find a rank-256 solution among optimal solutions of the SDP relaxation.

the table shows that the SDP relaxation was unable to find an orthogonal solution. This is because of the ε-rank of the solution V that is higher than the desired value $m = 256$.

Since we were tasked with finding an orthogonal solution, we used the modified log-det heuristic (Algorithm 1 with (70)) to search for a rank-m solution among the optimal solutions of the SDP relaxation. As shown in Table 16, the modified log-det heuristic was successful in finding such a solution. This solution provided the same optimal value as the SDP relaxation, which was even slightly lower than the one yielded by the solution of the OLSR algorithm from [107]. However, the orthogonality criterion was a bit less accurate, which could be improved by choosing a stricter value of ε for the stopping criterion of Algorithm 1.

As for the computation time, the SDP approach cannot compete with the OLSR algorithm [107], as its computation time is more than 1000 times worse, which is caused by the fact that the SDP approach relies on solving SDP problems of large size ($p = 165$, $m = 256$, and $n = 15$). On the other hand, the Frobenius norm is sensitive to outliers and hence the l_1 norm is more suitable for this kind of application, as stated in [107]. However, the available methods for solving (9) with the l_1 norm in the objective are limited to those

based on differential equations, as indicated in Table 11. Therefore, the proposed conic approach can be applied there and even its computation time for solving (9) with the l_1 norm is much better since handling a lower number of variables in reformulation (93).

5.1.3.4 Standard unbalanced OPPs with the Frobenius norm in the objective

In this set of experiments, we focus on solving standard unbalanced OPPs with the Frobenius norm in the objective. This subclass of OPPs is chosen to enable a comparison of the proposed rank-constrained SDP reformulation (92) with existing approaches developed specifically for this type of OPPs. Namely, we use the OLSR algorithm from [107] that was employed also in the previous subsection, and the SDP relaxation of the form (90) proposed in [28], which was discussed in Subsection 5.1.1.

Table 17 compares the results obtained by the existing methods and the proposed conic approach in solving standard unbalanced OPPs of the form (85) with the Frobenius norm in the objective, where $B = I_n$, $m = 5$, $n = 3$, $p = 30$, $q = n$. The problems for the first two columns were generated with a zero optimal value ($f^* = 0$), and the problems for the last two columns were generated with a nonzero optimal value ($f^* \neq 0$).

The results presented in Table 17 demonstrate that the OLSR algorithm [107] is highly efficient and consistently produces solutions that satisfy the orthogonality criterion with a high precision. However, the optimal value obtained by this method is generally higher than the ones obtained by the other methods. On the other hand, the SDP relaxation (90) from [28] yielded slightly better numerical results than the SDP relaxation of the proposed reformulation (92), as evidenced by the average rank of V, the average values of the orthogonality criterion, and the values of empirical ε. However, the advantage of the proposed conic approach lies in its ability to enhance a solution of the SDP relaxation by reducing its rank. To achieve this, we applied the proposed bisection algorithm (Algorithm 4) in its log-det and convex iteration versions. Both versions succeeded in finding a rank-m solution, although it came at the expense of computation time, as demonstrated by results in Table 17.

In summary, the OLSR algorithm [107] is the best option for finding a feasible solution to the standard unbalanced OPP. If one requires an optimal solution, the SDP relaxation (90) from [28] provides a good balance between solution quality and computation time.

method	criterion	$f^* = 0$		$f^* \neq 0$	
		(10,3,20)	(15,5,20)	(10,3,20)	(15,5,20)
OLSR algorithm from [107]	$\|C - AX\|_F$	4.2345e-01	1.1607e-01	7.3917	11.6977
	$\|X^T X - I_n\|_F$	1.3581e-15	2.0473e-15	1.5734e-15	2.1533e-15
	$rank(V)$	10	15	10	15
	time (s)	0.0015	0.0034	0.0033	0.0057
	empirical ε	1.3771e-15	1.6864e-15	1.5477e-15	1.5603e-15
SDP relaxation (90) from [28]	$\|C - AX\|_F$	2.6592e-04	3.1659e-04	3.0475	3.1981
	$\|X^T X - I_n\|_F$	1.1943e-08	2.3035e-07	4.0761e-08	1.6570e-03
	$rank(V)$	10	15.02	10.90	15.18
	time (s)	0.2063	0.2184	0.2501	0.2690
	empirical ε	9.8172e-09	1.3052e-06	3.8028e-07	3.6943e-02
proposed SDP relaxation of (92)	$\|C - AX\|_F$	2.1963e-05	2.0107e-05	2.9164	2.9757
	$\|X^T X - I_n\|_F$	1.5452e-06	2.1261e-06	5.5630e-02	1.0713e-01
	$rank(V)$	10.05	15.21	12.42	17.26
	time (s)	0.3623	0.5448	0.4047	0.5090
	empirical ε	1.2587e-06	2.7242e-06	6.7747e-02	1.2386e-01
bisection alg. (log-det heu.) for (92)	$\|C - AX\|_F$	1.4094e-07	8.1656e-08	3.0212	3.2049
	$\|X^T X - I_n\|_F$	1.4094e-07	8.1656e-08	1.6072e-06	1.2667e-06
	$rank(V)$	10	15	10	15
	time (s)	34.54	51.43	54.28	85.95
bisection alg. (convex iter.) for (92)	$\|C - AX\|_F$	2.9968e-06	3.2103e-07	2.9876	3.2053
	$\|X^T X - I_n\|_F$	1.0383e-06	8.6009e-07	1.9979e-06	1.9879e-06
	$rank(V)$	10	15	10	15
	time (s)	132.36	159.98	123.29	164.33

Table 17: Comparison of the existing methods and the proposed conic approach in solving standard unbalanced OPPs with the Frobenius norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, computation time and maximum value of empirical ε obtained by the OLSR algorithm [107], the SDP relaxation [28], the proposed SDP relaxation of (92) and the bisection algorithm (Algorithm 4). Averages counted for 50 generated problems of size (m, n, p) with optimal value f^*.

If computation time is not a concern, the proposed bisection algorithm (Algorithm 4) applied to solve the rank-constrained SDP reformulation of the unbalanced OPP in the form (92) can provide an orthogonal solution with a lower objective value than other approaches, that is, a better approximation of the optimal solution.

5.1.3.5 Weighted OPPs with the Frobenius norm, l_1 norm, l_∞ norm and spectral norm in the objective

This subsection focuses on weighted OPPs of the form (85), where A and B are general matrices not equal to the identity. In order to demonstrate the versatility of the proposed conic approach, we applied it to solve weighted OPPs that use different types of matrix norms to define the objective of (85). In the following sets of experiments, we focus on weighted OPPs with the parameters $p = 10$, $m = 4$, $n = 4$, and $q = 3$. For each of the considered matrix norms, including the Frobenius norm, l_1 norm, l_∞ norm, and spectral norm, we generated 50 problems with a zero optimal value ($f^* = 0$) and 50 problems with a nonzero optimal value ($f^* \neq 0$), as described in the introduction to numerical results.

It is important to note that the methods discussed in the previous subsection cannot be utilized in this case, as they rely on the specific structure of unbalanced problems where B equals identity and the definition of the Frobenius norm. As displayed in Table 11, if the objective of the weighted OPP is defined in terms of Frobenius norm, they can be solved by several algorithms derived on the Stiefel manifold, such as the spectral projected gradient method [96, 41], which was experimentally demonstrated to be very effective. For the case of l_1 norm in the objective, a differential approach was proposed in [97]. However, the performance of such an approach was illustrated only on small examples, and the authors labeled this approach to be time-consuming since using built-in Matlab functions for ODE calculations. Regarding the l_2 norm and l_∞ norm in the objective of (85), there are no significant results in the literature (compare to Table 11).

Table 18 summarizes the results obtained by applying the conic approach to solve weighted OPPs (85) with the Frobenius norm in the objective. The first part of the table shows the performance of the SDP relaxation of the rank-constrained SDP reformulation (92). According to the orthogonality criterion, none of the 100 generated problems achieved a rank-4 solution. Therefore, rank reduction algorithms were applied, the results

method	criterion	(m,n,p,q) = (10,4,4,3)		
		$f^* = 0$	$f^* \neq 0$	
SDP relaxation	$\|C - AXB\|_F$	2.4501e-05	2.2556	
	$\|X^T X - I_n\|_F$	1.0000	1.0835	
	$rank(V)$	7.96	6.28	
	time (s)	0.3188	0.2642	
		(γ)	(α)	bisection
log-det heuristic	$\|C - AXB\|_F$	9.8639e-06	2.4110	2.3726
	$\|X^T X - I_n\|_F$	1.6141e-07	4.4423e-07	1.9947e-06
	$rank(V)$	4	4	4
	time (s)	1.0890	0.8495	47.5041
convex iteration	$\|C - AXB\|_F$	8.6404e-07	2.3420	2.3379
	$\|X^T X - I_n\|_F$	1.9869e-07	1.2403e-07	1.4682e-06
	$rank(V)$	4	4	4
	time (s)	1.4467	1.2530	93.9932

Table 18: Solving weighted OPPs with the Frobenius norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, and computation time obtained by the SDP relaxation of (92), the modified versions of algorithms, labeled (γ), the bi-criterion versions of algorithms, labeled as (α) and the bisection algorithm (Algorithm 4) in solving 100 generated weighted OPPs with the Frobenius norm in the objective of the size (m, n, p, q) with optimal value f^*.

of which are presented in the second part of the table. For problems with a zero optimal value, both the modified versions of the log-det heuristic (Algorithm 1 with (70)) and the convex iteration (Algorithm 3 with (71)) successfully found rank-4 solutions among the optimal solutions of the SDP relaxation for all generated cases.

On the other hand, when solving problems with a nonzero optimal value, the modified versions of the algorithms did not produce solutions of the desired rank. Therefore, we applied the bi-criterion versions of the log-det heuristic (Algorithm 1 with (65)) and the convex iteration (Algorithm 3 with (66)) with $\alpha = 10$, results of which are seen in column (α) in Table 18. The obtained rank-4 solutions served as inputs for the bi-

method	criterion	(m,n,p,q)=(10,4,4,3)			
		$f^* = 0$	$f^* \neq 0$		
SDP relaxation	$\|C - AXB\|_1$	6.8561e-10	3.6718 (3.9190/3.5203)		
	$\|X^T X - I_n\|_F$	1.0000	1.0865 (1.0519/1.1078)		
	$rank(V)$	5	7.02 (6.84/7.12)		
	time (s)	0.2306	0.2489 (0.2464/0.2505)		
		(γ) (100%)	(γ) (38%)	(α) (62%)	bisection (62%)
log-det heuristic	$\|C - AXB\|_1$	3.9190	3.7857	4.3477	3.8258
	$\|X^T X - I_n\|_F$	9.0840e-08	1.0897e-07	8.1745e-08	1.4246e-06
	$rank(V)$	4	4	4	4
	time (s)	0.7324	0.7364	0.6843	40.6750
convex iteration	$\|C - AXB\|_1$	4.9138e-10	3.9190	3.9067	3.7817
	$\|X^T X - I_n\|_F$	2.2343e-09	5.6597e-09	2.3512e-09	1.4875e-06
	$rank(V)$	4	4	4	4
	time (s)	0.9686	1.0110	1.1724	84.4602

Table 19: Solving weighted OPPs with the l_1 norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, and computation time obtained by the SDP relaxation of (95), the modified versions of algorithms, labeled (γ), the bi-criterion versions of algorithms, labeled as (α) and the bisection algorithm (Algorithm 4) in solving 100 generated weighted OPPs with the l_1 norm in the objective of size (m, n, p, q) with optimal value f^*.

section algorithm (Algorithm 4), which provided solutions of the rank-constrained SDP reformulation (92) providing slightly better objective value than the bi-criterion methods themselves.

An analogical procedure was used also to solve weighted OPPs (85) with the l_1 norm in the objective. Results of the executed experiments are summarized in Table 19. Similar to the previous experiments with the Frobenius norm, the SDP relaxation of the rank-constrained SDP reformulation (93) failed to find an orthogonal solution. However, the modified log-det heuristic (Algorithm 1 with (70)) and the modified convex iteration (Algorithm 3 with (71)) succeeded in producing a rank-4 optimal solution of the SDP

method	criterion	(m,n,p,q)=(10,4,4,3)			
		$f^* = 0$	$f^* \neq 0$		
SDP relaxation	$\|C - AXB\|_2$	1.8778e-09	1.7020 (1.7020/1.7020)		
	$\|X^T X - I_n\|_F$	1.0000	1.1803 (1.1783/1.1823)		
	$rank(V)$	5	7.38 (7/7.76)		
	time (s)	0.2360	0.2807 (0.2977/0.2637)		
		(γ) (100%)	(γ) (50%)	(α) (50%)	bisection (50%)
log-det heuristic	$\|C - AXB\|_2$	8.6712e-10	1.7020	1.8577	1.7955
	$\|X^T X - I_n\|_F$	6.4882e-08	3.6070e-08	7.4676e-08	1.4419e-06
	$rank(V)$	4	4	4	4
	time (s)	0.7805	0.7837	0.7072	32.6941
convex iteration	$\|C - AXB\|_2$	7.2729e-11	1.7020	1.8212	1.7951
	$\|X^T X - I_n\|_F$	2.9138e-08	6.4836e-09	7.9518e-09	1.5579e-06
	$rank(V)$	4	4	4	4
	time (s)	1.0054	1.0261	1.1851	76.4913

Table 20: Solving weighted OPPs with the spectral norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, and computation time obtained by the SDP relaxation of (95), the modified versions of algorithms, labeled (γ), the bi-criterion versions of algorithms, labeled as (α) and the bisection algorithm (Algorithm 4) in solving 100 generated weighted OPPs with the spectral norm in the objective of size (m, n, p, q) with optimal value f^*.

relaxation for all generated problems with a zero optimal value. In contrast to the previous experiments, the modified versions of the algorithms provided a rank-4 solution also in 38% of the generated problems with nonzero optimal value, as seen in column (γ). For the remaining 68% of problems, we used the bi-criterion versions of the algorithms to initialize the bisection algorithm (Algorithm 4) with the obtained rank-4 solutions. As a result, we obtained orthogonal solutions yielding a lower objective value than the bi-criterion versions of the algorithms (see columns (α) and bisection).

Table 20 presents the results for weighted OPPs with the spectral norm in the objective. In this case, the rank-constrained SDP reformulation takes the form (95). Similar

method	criterion	(m,n,p,q)=(10,4,4,3)			
		$f^* = 0$	$f^* \neq 0$		
SDP relaxation	$\|C - AXB\|_2$	8.2965e-10	1.3773 (1.4607/1.3614)		
	$\|X^T X - I_n\|_F$	1.0000	1.1161 (1.0582/1.1271)		
	$rank(V)$	5	6.76 (6.38/6.83)		
	time (s)	0.2498	0.2421 (0.2362/0.2432)		
		(γ) (100%)	(γ) (16%)	(α) (84%)	bisection (84%)
log-det heuristic	$\|C - AXB\|_2$	2.5135e-9	1.4607	1.8713	1.4894
	$\|X^T X - I_n\|_F$	1.2462e-8	5.3399e-7	6.1945e-8	1.5966e-6
	$rank(V)$	4	4	4	4
	time (s)	0.7022	0.7158	0.6863	38.0758
convex iteration	$\|C - AXB\|_2$	5.3009e-10	1.4711	1.5983	1.4719
	$\|X^T X - I_n\|_F$	2.0831e-9	2.3399e-9	1.9335e-9	1.1707e-6
	$rank(V)$	4	4	4	4
	time (s)	0.9691	0.9559	0.9473	72.5916

Table 21: Solving weighted OPPs with the l_∞ norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, and computation time obtained by the SDP relaxation of (94), the modified versions of algorithms, labeled (γ), the bi-criterion versions of algorithms, labeled as (α) and the bisection algorithm (Algorithm 4) in solving 100 generated weighted OPPs with the l_∞ norm in the objective of size (m, n, p, q) with optimal value f^*.

to the previous cases, the SDP relaxation did not provide an orthogonal solution and the modified log-det heuristic (Algorithm 1 with (70)) and the modified convex iteration (Algorithm 3 with (71)) were able to find an orthogonal solution among the optimal solutions of the SDP relaxation of (95) for all 50 generated problems with a zero optimal value, within a certain tolerance. For problems with a nonzero optimal value, the modified versions of the algorithms yielded rank-4 solutions for 50% of the generated problems, the results of which are summarized in column (γ). For the other 50% of problems, rank-4 solutions were obtained by the bi-criterion versions of the algorithms (see column (α)), which were then enhanced by the bisection algorithm (Algorithm 4).

The results for solving the weighted OPP (85) with the l_∞ norm in the objective are presented in Table 21 and can be interpreted similarly to the previous cases. In this case, we used the rank-constrained SDP reformulation of the form (94).

To summarize, the results presented in Table 18, Table 19, Table 20, and Table 21 demonstrate the performance of the proposed conic approach in solving weighted OPPs with various types of matrix norms in the objective. The results revealed that the modified version of the log-det algorithm (Algorithm 1 with (70)) and the modified version of the convex iteration (Algorithm 3 with (71)) were successful in finding low-rank solutions among optimal solutions of the SDP relaxation in cases where the optimal value was zero. This indicates that the conic approach may be suitable for finding orthogonal matrices that satisfy a linear system of equations. Since the bi-criterion log-det algorithm (Algorithm 1 with (65)) and the bi-criterion version of the convex iteration (Algorithm 3 with (66)) provided a tight interval for the bisection algorithm (Algorithm 4), we observe only slight improvements of the optimal value. We are aware that the computation time cannot compete with algorithms for solving weighted OPPs with the Frobenius norm. However, it can serve as an alternative to the differential approach [97], which is considered to be also time-consuming and what is the most important, it provides a tool for solving weighted OPPs with the l_2 norm and l_∞ norm, which are not covered by the existing approaches.

5.1.3.6 Balanced OPPs with additional linear constraints

As mentioned in the introduction of this chapter, the proposed conic approach is applicable to Procrustes problems with additional linear constraints presented in the problem formulation (85). An interesting example of such a problem is finding a permutation matrix that minimizes the objective of the standard balanced OPP (85), involving the Frobenius norm or the l_1 norm. A permutation matrix is a doubly stochastic matrix having nonnegative binary entries (0 or 1) with rows and columns summing to 1. Although the problem of finding such a matrix is an integer program, we use that a permutation matrix can be represented as an orthogonal matrix with nonnegative elements. Therefore, the standard balanced OPP with additional linear constraints representing finding

a permutation matrix can be formulated as follows

$$\begin{aligned} \min_{X \in \mathbb{R}^{n \times n}} \quad & \|C - AX\| & & \\ & X^T X &= I_n & \\ & X_{ij} &\geq 0, \qquad & \forall i, j = 1, ..., n. \end{aligned} \tag{96}$$

In the following set of experiments, we solve the OPP with the additional linear constraints of the form (96) generated for a random permutation matrix of size n. We focus on problems with a zero optimal value, which enables interpreting (96) as the problem of finding a permutation matrix that satisfies a linear system of equations.

Table 22 presents the results for using the Frobenius norm, while Table 23 shows the results for using the l_1 norm in the objective of (96). It is worth noting that the additional linear constraints can be easily incorporated into the rank-constrained SDP reformulation (92) for the Frobenius norm and (93) for the l_1 norm. In the following experiments, we generated problems of three different sizes with zero optimal values and solved them using the SDP relaxation of (92) and (93), respectively. Table 22 shows the optimality and orthogonality of the obtained solutions. For the l_1 norm, a rank-n solution was obtained in all cases. It was observed that even in cases where a rank-n solution was not obtained, the values of the empirical ε and the orthogonality criterion suggest that the lack of a rank-n solution was due to the strict value of ε.

We test the feasibility of the solutions of (96) using several criteria. The first pair of criteria, $\|X\mathbf{1}_n - \mathbf{1}_n\|_1$ and $\|X^T\mathbf{1}_n - \mathbf{1}_n\|_1$, verifies whether the row and column sums of the solution matrix X equal 1. The third criterion, $\|o_{max} - \mathbf{1}n\|_1$, where o_{max} is a vector of the n largest elements of X, checks if the solution has exactly n elements equal to 1. Finally, the fourth criterion, $\|z_{min} - \mathbf{0}_{n(n-1)}\|_1$, where z_{min} is the vector of $n(n-1)$ smallest elements of X, verifies if the solution has exactly $n(n-1)$ elements equal to 0.

To conclude, the proposed conic approach was successful in solving OPPs with additional linear constraints. Moreover, it was observed that the l_1 norm is more effective in handling matrices with many zero elements, which is the case of permutation matrices.

criterion \ (m,n,p)	(3,3,10)	(5,5,20)	(10,10,30)
$\|C - AX\|_F$	3.3660e-5	4.9711e-5	5.2977e-10
$\|X^T X - I_n\|_F$	2.6449e-6	2.7031e-6	2.3347e-11
$rank(V)$	3.59	5.49	10
# of $rank(V) > m$	52	14	0
$\|X\mathbf{1}_n - \mathbf{1}_n\|_1$	1.7330e-9	1.1840e-8	3.8232e-13
$\|X^T\mathbf{1}_n - \mathbf{1}_n\|_1$	2.8542e-9	1.1274e-8	1.8038e-13
$\|o_{max} - \mathbf{1}_n\|_1$	1.8478e-6	2.6850e-6	3.4809e-11
$\|z_{min} - \mathbf{0}_{n(n-1)}\|_1$	1.8462e-6	2.6831e-6	3.4809e-11
time (s)	0.2150	0.3743	1.1181
empirical ε	2.0240e-6	2.0534e-6	1.4831e-11

Table 22: Solving standard balanced OPPs with the Frobenius norm in the objective and additional linear constraints. Results obtained by the SDP relaxation in solving 100 generated OPPs representing problems of finding permutation matrices minimizing the objective.

criterion \ (m,n,p)	(3,3,10)	(5,5,20)	(10,10,30)
$\|C - AX\|_1$	3.0607e-10	4.4304e-10	5.2880e-10
$\|X^T X - I_n\|_F$	2.3621e-11	2.4869e-11	2.3348e-11
$rank(V)$	3	5	10
# of $rank(V) > m$	0	0	0
$\|X\mathbf{1}_n - \mathbf{1}_n\|_1$	4.1927e-12	7.6000e-13	3.8232e-13
$\|X^T\mathbf{1}_n - \mathbf{1}_n\|_1$	4.1299r-12	8.3364e-13	1.8042e-13
$\|o_{max} - \mathbf{1}_n\|_1$	1.4740e-11	2.4448e-11	3.4865e-11
$\|z_{min} - \mathbf{0}_{n(n-1)}\|_1$	1.8849e-11	2.5154e-11	3.4811e-11
time (s)	0.1598	0.2892	0.5307
empirical ε	3.6912e-11	2.2944e-11	1.4833e-11

Table 23: Solving standard balanced OPPs with the l_1 norm and additional linear constraints. Results obtained by the SDP relaxation in solving 100 generated OPPs representing problems of finding permutation matrices minimizing the objective.

5.1.3.7 Extension - Graph isomorphism problem as a two-sided OPP

This subsection describes the graph isomorphism problem and its formulation as a special type of OPP, known as a two-sided OPP (see [54]). We also introduce the proposed conic approach to solve this problem and provide solutions for four different graph isomorphism problems of different sizes.

It is known that a graph is a set of vertices connected by edges. Determining if two graphs are isomorphic or not is important in various fields such as chemistry, computer science, and data mining. For the sake of simplicity, we only consider unweighted undirected graphs.

Definition 5.1 ([48]). *An isomorphism of two graphs G and $\tilde{G}$ is a bijection between the vertex sets of G and $\tilde{G}$:*

$$f\colon V(G) \to V(\tilde{G}) \tag{97}$$

such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in $\tilde{G}$.

To put it simply, according to Definition 5.1, two graphs are isomorphic if the vertices are tied to edges, and we can obtain the second graph only by moving the vertices of the first graph, as shown in Figure 13.

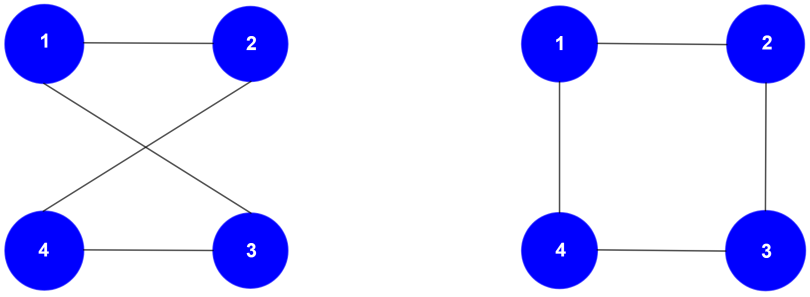


Figure 13: Example of isomorphic graphs.

We can label the n vertices of a simple graph as $1, 2, ..., n$. The graph can then be defined by its adjacency matrix $A \in \mathbb{R}^{n\times n}$, where each element $A_{ij} \in \{0, 1\}$ indicates whether vertices i and j are adjacent or not. This definition can be extended to weighted graphs as well. Graph isomorphism can be determined using the adjacency matrices as stated in Definition 5.2.

Definition 5.2 ([92]). *Graphs G and $\tilde{G}$ are isomorphic if and only if there is a permutation matrix $P \in \mathbb{R}^{n\times n}$ that satisfies*

$$PA = \tilde{A}P. \tag{98}$$

It means that an isomorphic graph is created by permuting rows and columns of the adjacency matrix of the original graph in the same order. Note that Definition 5.2 is equivalent to Definition 5.1.

As in [92], consider two finite unweigted undirected graphs with n vertices labeled $1, 2, ..., n$ and described by their adjacency matrices $A, \tilde{A} \in \mathbb{S}^n$. The task of the graph isomorphism problem is to determine whether the two given graphs are isomorphic, meaning that there exists a permutation that satisfies (98), and to find such a permutation $P \in \mathbb{R}^{n\times n}$ if it exists. Unlike [92], we do not represent a permutation matrix as a matrix having unit sums of rows and columns and elements equal to 0 or 1 since such a problem is an integer problem where relaxations need to be applied leading to a solution having non-integer elements and rounding algorithms have to be tailored to get a permutation matrix. To avoid this, we rather represent the permutation matrix as an orthogonal matrix having nonnegative elements. Consequently, the graph isomorphism problem can be formulated as a so-called two-sided OPP (see [54]) of the following form

$$\begin{aligned} \min_{P\in\mathbb{R}^{n\times n}} \quad & \|PA - \tilde{A}P\|_1 \\ & P^TP = I_n \\ & P_{ij} \geq 0 \qquad \forall i, j = 1, ..., n. \end{aligned} \tag{99}$$

In [40], the graph isomorphism problem (99) is formulated with the Frobenius norm in the objective. However, as demonstrated in the previous subsection, the l_1 norm is more suitable for handling matrices with a large number of zero elements, therefore it denotes our preference. The constraints of (99) define the set of permutation matrices $P \in \mathbb{R}^{n\times n}$.

Note that the objective function of (99) fits the formulation of a general norm minimization problem (21), therefore it can be rewritten according to Proposition 2.2. Furthermore, the first constraint of (99) is the orthogonality constraint, which can be rewritten as in (86). By implementing these modifications, we can obtain the following rank-

constrained SDP reformulation

$$\begin{aligned}
\min_{P\in\mathbb{R}^{n\times n},t\in\mathbb{R},S\in\mathbb{R}^{n\times n},V\in\mathbb{S}^{2n}} \quad & t \\
-S \leq \quad PA-\tilde{A}P &\leq S, \\
S^T\mathbf{1}_n &\leq t\mathbf{1}_n, \\
P_{ij} &\geq 0, \qquad \forall i,j=1,...,n. \\
V=\begin{bmatrix} I_n & P^T \\ P & I_n \end{bmatrix} &\succeq 0, \\
rank(V) &= n.
\end{aligned} \tag{100}$$

The problem (100) can be tackled using the techniques presented in Section 2. Furthermore, we can exploit the fact that two graphs represented by their adjacency matrices A and $\tilde{A}$ are isomorphic if and only if P is a feasible solution of (99) with a zero optimal value. Therefore, in our numerical experiments, we solve the rank-constrained SDP problem (100) using the SDP relaxation, the modified versions of the log-det heuristic (Algorithm 1 with (70)), and the convex iteration (Algorithm 3 with (71)), where γ is a small positive constant. We use four pairs of adjacency matrices of different sizes $n \in \{4, 6, 16, 25\}$ that represent pairs of isomorphic graphs, and we search for a permutation matrix that satisfies (98). The results are summarized in Table 24.

The results in Table 24 demonstrate that the SDP relaxation of (100) provided optimal solutions of higher rank than expected, with optimal values greater than zero. In contrast, the modified log-det heuristic and the modified convex iteration produced solutions for (100) that satisfy the conditions listed in the table, which confirms both their feasibility and ε-optimality. Since these solutions are permutation matrices, they can be considered optimal solutions for the graph isomorphism problem (99).

5.2 Oblique Procrustes problems

Procrustes problems defined over a feasible set given by quadratic constraints of the form $diag(X^TX) = \mathbf{1}_n$ are referred to as the *oblique Procrustes problems* (ObPPs). It is common to consider the Frobenius norm (see [54, 96]) or the l_1 norm in the objective (see [98, 14]). ObPPs arise in various applications such as factor analysis [76] and shape analysis [30]. The ObPPs with a partially specified target and weighted ObPPs are also discussed in [98].

method	criterion	n=4	n=6	n=16	n=25
SDP relaxation	$\|PA - \tilde{A}P\|_1$	2	3.5187	6.0017	0.8683
	$\|P^TP - I_n\|_F$	1.7321	1.5643	2.4076	3.7997
	$rank(V)$	7	11	31	49
	$\|P\mathbf{1}_n - \mathbf{1}_n\|_1$	7.2384e-11	1.4627e-11	1.1875e-10	2.0370e-12
	$\|P^T\mathbf{1}_n - \mathbf{1}_n\|_1$	7.2384e-11	1.4627e-11	1.1875e-10	2.0367e-12
	$\|o_{max} - \mathbf{1}_n\|_1$	3	2.6188	5.8639	13.8898
	$\|z_{min} - \mathbf{0}_{n(n-1)}\|_1$	3	2.6188	5.8639	13.8898
	time (s)	0.2473	0.2827	0.8153	1.9829
	empirical ε	0.4896	0.5958	1.6218	3.9723
modified log-det heuristic ($\gamma = \varepsilon$)	$\|PA - \tilde{A}P\|_1$	1.5560e-15	4.2576e-13	1.2472e-13	7.6111e-7
	$\|P^TP - I_n\|_F$	8.7483e-12	1.4113e-7	1.2744e-7	6.1667e-6
	$rank(V)$	4	6	16	25
	$\|P\mathbf{1}_n - \mathbf{1}_n\|_1$	7.0166e-13	1.3124e-12	5.5889e-10	9.6092e-9
	$\|P^T\mathbf{1}_n - \mathbf{1}_n\|_1$	7.0166e-13	1.3096e-12	5.5889e-10	9.6092e-9
	$\|o_{max} - \mathbf{1}_n\|_1$	7.3948e-12	7.0571e-8	2.072e-7	1.4358e-5
	$\|z_{min} - \mathbf{0}_{n(n-1)}\|_1$	8.0964e-12	7.0570e-8	2.0716e-7	1.4348e-5
	time (s)	0.4896	0.5958	1.6218	3.9723
	empirical ε	2.5248e-12	7.0564e-8	2.5362e-8	9.3755e-7
modified convex iteration ($\gamma = \varepsilon$)	$\|PA - \tilde{A}P\|_1$	3.7889e-16	2.4484e-14	7.1029e-15	4.6085e-15
	$\|P^TP - I_n\|_F$	5.1683e-8	1.2035e-9	4.4462e-9	4.4291e-8
	$rank(V)$	4	6	16	25
	$\|P\mathbf{1}_n - \mathbf{1}_n\|_1$	4.6035e-11	1.7152e-12	1.9921e-10	1.0913e-9
	$\|P^T\mathbf{1}_n - \mathbf{1}_n\|_1$	4.6036e-11	1.7154e-12	1.9921e-10	1.0913e-9
	$\|o_{max} - \mathbf{1}_n\|_1$	4.4882e-9	1.0254e-9	8.5908e-9	1.0850e-7
	$\|z_{min} - \mathbf{0}_{n(n-1)}\|_1$	4.4422e-9	1.0247e-9	8.3916e-9	1.0744e-7
	time (s)	0.5332	0.7158	3.4213	11.5446
	empirical ε	1.4922e-9	4.3997e-10	7.2807e-10	4.5204e-9

Table 24: Solving the graph isomorphism problem. Results obtained by the SDP relaxation, the modified log-det heuristic (Algorithm 1 with (70)) and the modified convex iteration (Algorithm 3 with (71)) for $\gamma = 10^{-6}$ in solving two-sided OPPs of the form (100).

In general, the ObPP can be formulated as follows

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}} \quad & f(X) := \|W \circ (C - AXB)\| \\ & diag(X^T X) = \mathbf{1}_n. \end{aligned} \tag{101}$$

Solution algorithms for a standard ObPP, where B is an identity, are based on the appealing feature that the objective and constraint are separable with respect to columns of the matrix variable $X \in \mathbb{R}^{m\times n}$. In [98] and [14], the proposed methods separate the problem (101) into n problems. The i-th problem (for $i = 1, .., n$) has the form

$$\begin{aligned} \min_{X_i\in\mathbb{R}^m} \quad & \|W_i \circ (C_i - A_i^T X_i)\| \\ & X_i^T X_i = 1, \end{aligned} \tag{102}$$

where W_i, C_i, X_i are i-th columns of matrices $W \in \mathbb{R}^{p\times n}, C \in \mathbb{R}^{p\times n}, X \in \mathbb{R}^{m\times n}$ and A_i^T is an i-th row of $A \in \mathbb{R}^{p\times m}$. Consequently, a smooth reformulation is applied to each problem (102), and they are solved by standard nonlinear programming methods. However, this approach only covers problems with the l_1 norm or the Frobenius norm in the objective.

5.2.1 The proposed conic approach

In this part, the conic reformulation of the ObPP (101) is presented. The quadratic constraint $diag(X^T X) = \mathbf{1}_n$ can be rewritten using its representation from Table 2 to obtain the following problem

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}, G\in\mathbb{S}^n, V\in\mathbb{S}^{m+n}} \quad & f(X) := \|W \circ (C - AXB)\| \\ & diag(G) = \mathbf{1}_n, \\ & V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} \succeq 0, \\ & rank(V) = m. \end{aligned} \tag{103}$$

If the objective of (103) contains l_1 norm, we apply Proposition 2.2 to get a rank-

constrained SDP reformulation in the form

$$\begin{aligned}\min_{X\in\mathbb{R}^{m\times n},G\in\mathbb{S}^n,V\in\mathbb{S}^{m+n},S\in\mathbb{R}^{p\times q},t\in\mathbb{R}} \quad & t \\ diag(G) &= \mathbf{1}_n, \\ V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} &\succeq 0, \\ rank(V) &= m, \\ -S \leq W \circ (C - AXB) &\leq S, \\ S^T\mathbf{1}_p &\leq t\mathbf{1}_q.\end{aligned} \tag{104}$$

If the objective of (103) is defined in terms of the l_∞ norm, Proposition 2.3 can be applied to get a rank-constrained SDP problem analogical to (104) with the last constraint $S\mathbf{1}_q \leq t\mathbf{1}_p$.

If (103) uses the Frobenius norm, the statement of Proposition 2.5 is applied to obtain

$$\begin{aligned}\min_{X\in\mathbb{R}^{m\times n},G\in\mathbb{S}^n,V\in\mathbb{S}^{m+n},Z\in\mathbb{S}^p} \quad & tr(Z) \\ diag(G) &= \mathbf{1}_n, \\ V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} &\succeq 0, \\ rank(V) &= m, \\ \begin{pmatrix} I_q & (W \circ (C - AXB))^T \\ W \circ (C - AXB) & Z \end{pmatrix} &\succeq 0.\end{aligned} \tag{105}$$

In the case of the spectral norm in the objective of (103), Proposition 2.4 provides a rank-constrained reformulation of the form

$$\begin{aligned}\min_{X\in\mathbb{R}^{m\times n},G\in\mathbb{S}^n,V\in\mathbb{S}^{m+n},s\in\mathbb{R}} \quad & s \\ diag(G) &= \mathbf{1}_n, \\ V = \begin{pmatrix} I_m & X \\ X^T & G \end{pmatrix} &\succeq 0, \\ rank(V) &= m, \\ \begin{pmatrix} sI_p & W \circ (C - AXB) \\ (W \circ (C - AXB))^T & sI_q \end{pmatrix} &\succeq 0.\end{aligned} \tag{106}$$

5.2.2 Numerical results

This section provides an overview of the numerical results obtained from solving different types of oblique Procrustes problems (ObPPs). First, we solve standard balanced ObPPs with either the Frobenius or l_1 norm in the objective using separation methods based on solving n problems of the form (102), as introduced in [98, 14] and included in Table 11. Then, we compare our results with those obtained using a separation method to demonstrate the accuracy of the proposed conic approach. Subsequently, we focus on weighted ObPPs with various matrix norms in the objective, which include not only the Frobenius norm and the l_1 norm but also the l_∞ norm and the spectral norm.

In the following experiments, we generate problems with either a zero or nonzero optimal value. For both cases, we generate the oblique matrix using the method proposed in [96], which involves the following procedure: $X_{oblique} = XDiag(diag(X^TX)^{\frac{1}{2}})$, where $X \in \mathbb{R}^{m\times n}$ is a random matrix. To evaluate the feasibility of the obtained solution X, we use the criterion $\|diag(X^TX) - \mathbf{1}_n\|_1$. For all experiments, we apply the algorithms from Chapter 2 with the following inputs: $\varepsilon = 10^{-6}$, $\delta = 0.01$, $\rho = 10^{-6}$, and $M = 10$.

5.2.2.1 Standard oblique Procrustes problems

In this section, we deal with the standard ObPPs of the form (101) where B equals the identity. These problems have a separable structure, which allows us to compare the SDP relaxations of the rank-constrained SDP reformulations (105) and (104) with methods based on solving a sequence of n problems of the form (102).

Tables 25 and 26 summarize the results obtained for 100 generated ObPPs with the l_1 norm and the Frobenius norm in the objective, respectively. The results are obtained using the SDP relaxation and a separation method based on solving (102). It is worth noting that the rank of the block matrix V is connected to the criterion $\|diag(X^TX)-\mathbf{1}_n\|_1$ that checks whether the solution X is oblique.

In summary, the results in Tables 25 and 26 demonstrate that the conic approach is applicable to solve standard ObPPs. Its results are comparable to those of the separation methods, but with significantly shorter computation time, since the SDP relaxation solves only one optimization problem instead of n problems like in the separation method. This highlights the effectiveness of the conic approach in solving standard ObPPs.

method	criterion	$(10, 3, 20)$	$(15, 5, 30)$	$(20, 10, 30)$
semidefinite relaxation	$\|C - AX\|_1$	4.6713e-10	4.4794e-10	4.5795e-10
	$\|diag(X^T X) - \mathbf{1}_n\|_1$	2.4854e-11	2.7149e-11	7.0218e-11
	$rank(V)$	10	15	20
	time (s)	0.1870	0.2141	0.3065
separation method	$\|C - AX\|_1$	1.8003e-09	4.1755e-09	1.1093e-08
	$\|diag(X^T X) - \mathbf{1}_n\|_1$	2.0960e-11	3.7786e-11	1.2183e-10
	time (s)	0.5053	0.8450	1.6739

Table 25: Solving standard ObPPs with the l_1 norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, and computation time obtained by the SDP relaxation of (104) and the separation method (102) to solve 100 generated problems of size (m, n, p) with optimal value $f^* = 0$.

method	criterion	$(10, 3, 20)$	$(15, 5, 30)$	$(20, 10, 30)$
semidefinite relaxation	$\|C - AX\|_F$	2.4151e-10	2.0336e-10	3.0301e-10
	$\|diag(X^T X) - \mathbf{1}_n\|_1$	2.8618e-11	2.6972e-11	6.8173e-11
	$rank(V)$	10	15	20
	time (s)	0.1757	0.2111	0.6165
separation method	$\|C - AX\|_F$	7.1391e-10	9.9484e-10	2.4307e-09
	$\|diag(X^T X) - \mathbf{1}_n\|_1$	2.8311e-11	3.4651e-11	1.1039e-10
	time (s)	0.5161	0.8273	3.3633

Table 26: Solving standard ObPPs with the Frobenius norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, and computation time obtained by the SDP relaxation of (105) and the separation method (102) to solve 100 generated problems of size (m, n, p) with optimal value $f^* = 0$.

5.2.2.2 Weighted oblique Procrustes problems with the Frobenius norm, l_1 norm, l_∞ norm and spectral norm in the objective

When dealing with weighted ObPPs of the form (101), where A and B are not equal to the identity, the problem loses its separable structure, rendering separation methods

norm	criterion	SDP relaxation	log-det (α)	cvx.iter. (α)	bisection (log-det)	bisection (cvx.iter.)
Frob. norm	$\|C - AXB\|_F$	2.1410	2.2327	2.1658	2.1533	2.1658
	$\|diag(X^TX) - \mathbf{1}_n\|_1$	2.6696e-01	5.6746e-08	2.1883e-08	3.6455e-06	2.1883e-08
	$rank(V)$	5.45	4	4	4	4
	time (s)	0.2624	0.5002	0.9735	37.6552	21.5325
	%	10	100	100	100	100
l_1 norm	$\|C - AXB\|_1$	3.3923	3.9017	3.4994	3.4455	3.3415
	$\|diag(X^TX) - \mathbf{1}_n\|_1$	0.5372	9.6297e-09	2.8595e-09	5.2837e-07	5.2837e-07
	$rank(V)$	7.40	4	4	4	4
	time (s)	0.2318	0.4605	1.0019	14.9645	38.4616
	%	5	100	100	100	100
l_2 norm	$\|C - AXB\|_2$	1.6164	2.0091	1.8257	1.6246	1.6215
	$\|diag(X^TX) - \mathbf{1}_n\|_1$	1.1270	2.6523e-08	1.5957e-08	1.0289e-05	1.3706e-07
	$rank(V)$	7.7	4	4	4	4
	time (s)	0.2337	0.4630	0.9534	15.4821	36.2344
	%	0	100	100	100	100
l_∞ norm	$\|C - AXB\|_\infty$	4.9754	1.6356	1.3709	1.2757	1.3016
	$\|diag(X^TX) - \mathbf{1}_n\|_1$	4.9754	3.2818e-08	3.1357e-09	4.1862e-07	4.0443e-07
	$rank(V)$	6.7	4	4	4	4
	time (s)	0.2302	0.4459	0.9302	16.2491	36.9748
	%	15	100	100	100	100

Table 27: Solving weighted ObPPs with different matrix norm in the objective. Average values of optimal value, orthogonality criterion, rank of the block matrix V, computation time and percentage of solutions having ε-rank equal to $m = 4$ obtained by the SDP relaxations of (103), the bi-criterion versions of algorithms, labeled as (α), and the bisection algorithm (Algorithm 4) when solving 100 generated weighted ObPPs with the Frobenius norm, l_1 norm, ł$_\infty$ norm and spectral norm in the objective of size (m, n, p, q) with optimal value $f^* \neq 0$.

unsuitable. Nevertheless, the proposed conic approach can solve these weighted problems. Moreover, through the following series of experiments, we aim to illustrate the practicality of the proposed conic approach in solving ObPPs with different matrix norm choices, such as the Frobenius norm, l_1 norm, l_∞ norm, and spectral norm.

Table 27 presents a summary of the results obtained in solving generated ObPPs of the form (101), including the usage of the SDP relaxation of the rank-constrained SDP reformulations: (105) for the Frobenius norm, (104) for the l_1 norm, and (106) for the spectral norm. It is observed that in some cases, the SDP relaxation found a rank-4 solution. For the other cases, the bi-criterion versions of the log-det heuristic (Algorithm 1 with (65)) and the convex iteration (Algorithm 3 with (66)) were used to provide a rank-4 solution for initializing the bisection algorithm (Algorithm 4). In all cases, the optimal solutions obtained using the proposed bisection algorithm improved the initial solutions obtained using the bi-criterion algorithms. Additionally, for the l_1 norm, l_∞ norm, and spectral norm, 70-80% of the optimal solutions were found among the optimal solutions of the SDP relaxation. This indicates that the modified versions of the algorithms would be sufficient to find solutions for most of the generated problems.

To conclude, the proposed conic approach can be considered a tool for solving even weighted ObPPs (101) for any of the considered types of matrix norms.

5.3 Other types of Procrustes problems

5.3.1 Semidefinite Procrustes problems

A subclass of Procrustes problems defined over the cone of symmetric positive semidefinite matrices is known as the *semidefinite Procrustes problem* (SDPP). This problem has been studied in several works, such as [6], [50], and [62], where the authors have formulated necessary and sufficient conditions for the optimum and compared the performance of several numerical algorithms. For instance, in [6, Theorem 3.2], it has been shown that X^* is an optimal solution of the semidefinite Procrustes problem of the form

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & f(X) := \|C - AX\|_F \\ & X \succeq 0, \end{aligned} \tag{107}$$

if and only if $X^* \succeq 0$, $\nabla f(X^*) = 0$, and $\nabla f(X^*)X^* = 0$. In [62], an algorithm for solving the semidefinite Procrustes problem of the form (107) has been designed based on computing the optimality conditions using specific singular value decompositions.

The SDPP is recognized in numerous applications such as structural analysis [21], signal processing [90], and finance [73]. The problem of finding the nearest covariance

$p\backslash n$	5	10	15	20
25	2.4077e-04	5.8795e-04	1.5706e-06	1.6701e-06
50	2.7803e-04	6.2560e-04	6.7038e-03	1.4579e-01
75	4.9439e-04	1.0880e-03	3.0394e-03	1.1802e-02
100	3.2497e-04	1.5067e-03	7.0409e-03	8.5092e-03

Table 28: Accuracy of the optimal values of SDPPs with the Frobenius norm. Average values of the objective $\|C - AX\|_F$ obtained by the SDP relaxation applied to 100 generated semidefinite PPs with the Frobenius norm in the objective for different values of the parameters p and n.

matrix can be formulated as a special case of the SDPP (107) with $m = n = p$ and $A = I_m$. This problem is commonly encountered when the initial estimate of a covariance matrix is non-positive semidefinite, which is common e.g. in foreign exchange markets [39, 73]. Note that techniques designed for the problem of finding the nearest correlation matrix (see Section 4.1) can be applied to solve this special case.

5.3.1.1 The proposed conic approach

Since the SDPP (107) has only a semidefinite constraint, we can rewrite the objective using the statement of Proposition 2.5 into the form

$$\begin{aligned} \min_{X\in\mathbb{S}^n, Z\in\mathbb{S}^p} \quad & tr(Z) \\ & X \succeq 0, \\ & \begin{pmatrix} I_n & (C-AX)^T \\ C-AX & Z \end{pmatrix} \succeq 0. \end{aligned} \tag{108}$$

Note that besides the standard formulation of the SDPP, the conic approach can be applied also when using other types of matrix norms in the objective of (107). If we handle the l_1, l_∞ or l_2 norm, after applying Proposition 2.2, Proposition 2.3, or Proposition 2.4, respectively, we obtain reformulations (23), (25), or (27) where $\mathcal{P} = \{X \in \mathbb{S}^n \mid X \succeq 0\}$ and $B = I_n$. Indeed, regardless of the type of the matrix norm defining the objective, the SDPP (107) can be equivalently reformulated as an SDP problem and solved by interior point methods, implemented in solvers.

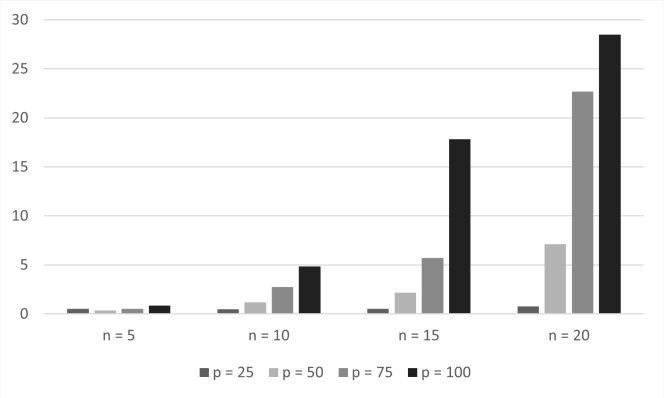


Figure 14: Accuracy of orthogonal solutions of SDPPs with the Frobenius norm. Average computation time (in seconds) of solving 100 generated semidefinite PPs with the Frobenius norm in the objective for different values of the parameters p and n.

5.3.1.2 Numerical results

In this section, we summarize the results obtained from solving SDPPs of the form (107). Since the reformulation (108) yields an SDP program, we directly used the IPMs implemented in the CVX modeling system [56], [55]. Table 28 shows the behavior of the average optimal value as the size of matrices $C \in \mathbb{R}^{p\times q}$ and $X \in \mathbb{S}^n$ increases. Additionally, Figure 14 provides information on the average computation time with respect to the parameters p and n.

5.3.2 Projection Procrustes problems

A projection Procrustes problem is formulated as follows

$$\begin{aligned} \min_{X\in\mathbb{S}^n} \quad & f(X) := \|W \circ (C - AXB)\| \\ & X^2 \;=\; X, \end{aligned} \tag{109}$$

where $W \in \mathbb{R}^{p\times q}, C \in \mathbb{R}^{p\times q}, A \in \mathbb{R}^{p\times n}$ and $B \in \mathbb{R}^{n\times q}$ are the given data.

A class of PPs defined over the set of projection matrices is not common in publications, although it can be used in various fields, such as computer vision, machine learning, and data analysis. For example, in [80], the k-means clustering algorithm was reformulated with the projection matrix variable. Often, also additional linear constraints on X are present in the problem formulation. This kind of problem arises e.g. in geometry [54].

5.3.2.1 The proposed conic approach

After applying Lemma 2.1, we can reformulate (109) as follows

$$\begin{aligned} \min_{X \in \mathbb{S}^n} \quad & f(X) := \|W \circ (C - AXB)\| \\ & V = \begin{pmatrix} I_n & X \\ X & X \end{pmatrix} \succeq 0, \\ & rank(V) = n. \end{aligned} \tag{110}$$

The objective of the projection PP (101) can be reformulated in a manner similar to the case of OPPs, as discussed in Subsection 5.1.2. Even when additional linear constraints are imposed on the projection matrix variable, the resulting projection PP can still be equivalently expressed as a rank-constrained SDP problem. Note that although we focus on the orthogonal projection matrix variable $X \in \mathbb{S}^n$, the proposed conic approach encompasses also problems with the non-orthogonal projection matrix variable $X \in \mathbb{R}^{m \times n}$ which is ensured by Theorem D.1.

5.3.2.2 Numerical results

Similarly to the previous subsection, we present the results in Table 29 and Table 30 to demonstrate that the SDP relaxation finds an optimal solution for each generated projection PP of the form (109), regarding different values of the parameters p and n. Furthermore, Figure 15 visually shows the increase in computation time as the values of parameters p and n increase.

$p \backslash n$	5	10	15	20
25	2.9454e-05	3.1622e-05	3.4590e-05	2.8262e-05
50	1.7348e-05	2.3380e-05	2.7099e-05	2.7412e-05
75	1.6140e-05	2.1091e-05	2.6705e-05	2.7428e-05
100	1.3999e-05	1.9272e-05	2.7378e-05	2.6619e-05

Table 29: Accuracy of the optimal values of projection PPs with the Frobenius norm in the objective. Average values of the objective $\|C - AX\|_F$ obtained by the SDP relaxation applied to 100 generated problem for different values of p and n.

$p\backslash n$	5	10	15	20
25	7.0839e-07	7.7668e-07	1.0709e-06	4.9143e-07
50	2.3967e-07	3.6361e-07	4.7086e-07	3.8615e-07
75	1.7934e-07	2.5984e-07	3.5800e-07	3.7077e-07
100	1.3548e-07	2.0522e-07	3.5962e-07	3.5351e-07

Table 30: Accuracy of projection criterion of projection PPs with the Frobenius norm in the objective. Average values of the criterion $\|X^2 - X\|_F$ obtained by the SDP relaxation applied to 100 generated problems for different values of p and n.

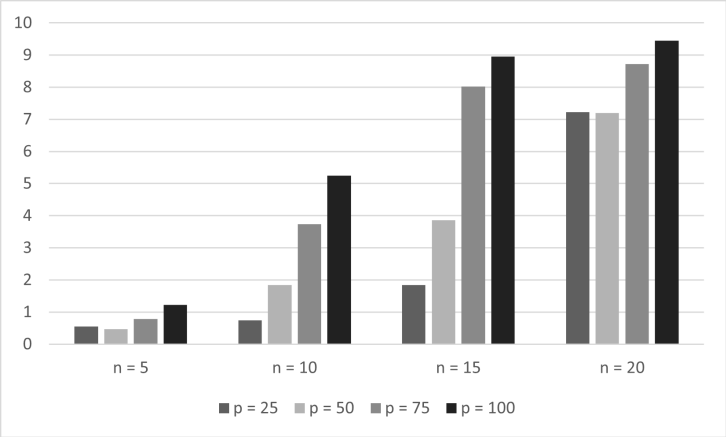


Figure 15: Computation time of projection PPs with the Frobenius norm in the objective. Average computation time (in seconds) of solving 100 generated problems for different values of p and n.

Conclusion

In this thesis, we proposed a conic optimization approach for solving matrix approximation problems in their generalized form (1). Our approach covers a broad class of problems as we consider minimization of an objective function defined in terms of different matrix norms, including l_1, l_2, l_∞, and Frobenius norm, over the feasibility set described by linear, semidefinite, quadratic, and rank constraints. Specifically, we studied standard matrix approximation problems that involve finding the nearest (low-rank) correlation matrix to the given empirical correlation matrix, and we also investigated various types of Procrustes problems including orthogonal, oblique, semidefinite, and projection matrix variables.

In Chapter 1, we introduced the generalized matrix approximation problem (1) and presented three applications that were solved in the final part of the thesis. In Chapter 2, we provided a comprehensive summary of known and lesser-known representations of quadratic constraints using linear, semidefinite, and rank constraints (see Section 2.2 and Table 2). We derived these representations using the Schur complement properties in Lemma 2.1, Proposition 2.1, and Appendix B.4. In Section 2.3, we investigated reformulations of norm minimization problems of the form (21), considering various matrix norms defining the objective. Proposition 2.2, Proposition 2.3, and Proposition 2.4 present known reformulations of the problem (21) using l_1, l_∞, and l_2 norms, respectively. Additionally, Proposition 2.5 proposed a new reformulation of the problem (21) with the Frobenius norm in the objective. We provided proofs for all the propositions and attached the detailed derivations in Appendix B.3. It has been shown that the generalized matrix approximation problem (1) can be equivalently reformulated as a semidefinite program with a possible rank constraint, as summarized in Subsection 2.4.

In Chapter 3, we provided an overview of well-known algorithms for solving rank-constrained optimization problems of the form (30). Since the convex relaxation (34) rarely finds a low-rank solution, we discussed other algorithms designed to solve the rank-constrained feasibility problem (36), such as the trace heuristic (39), the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3). To properly describe these algorithms, we outlined several auxiliary statements in the appendix, including a new

adaptation of the proof to Theorem C.1. Furthermore, we proposed a bisection algorithm (Algorithm 4) for solving the rank-constrained optimization problem (30). This algorithm guarantees to find an ε-optimal solution in a finite number of iterations, given an interval involving the optimal value and assuming the existence of reliable methods for solving feasibility problems of the form (68), as summarized and proved in Proposition 3.1. However, since we handle only the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3), which are not always guaranteed to converge to a solution of the desired rank, a solution found by the bisection algorithm (Algorithm 4) is considered only an approximation of the optimal solution to the rank-constrained optimization problem (30). Finally, we modified the log-det heuristic (Algorithm 1) and the convex iteration (Algorithm 3) to search for a low-rank solution among the optimal solutions of the convex relaxation (34), as explained in Section 3.3.3.

In Chapter 4, we demonstrated the applicability of the proposed conic optimization approach to solve real-life problems of finding the nearest correlation matrix (3) and the nearest low-rank correlation matrix (4) to the given empirical correlation matrix. We summarize the existing approaches for solving these problems in Section 4.1, and provide a (rank-constrained) SDP reformulation for both problems in the forms (78) and (80). In numerical experiments, we first solved Example 1.1 in Subsection 4.4.1 to illustrate the performance of the proposed approach, even in the case of the problem with the partially specified target. We then solved a set of generated problems of the form (3) to validate our results by comparing them with existing methods, as shown in Table 5. Additionally, we applied the proposed approach to rank-constrained problems of the form (4) to evaluate the performance of the bisection algorithm (Algorithm 4), which successfully found a good approximation of an optimal solution, as presented in Table 9.

In Chapter 5, we focus on solving various types of Procrustes problems and present the (rank-constrained) SDP reformulation of specific subclasses, including orthogonal, oblique, semidefinite and projection Procrustes problems. Table 11 illustrates the heterogeneity of methods designed specifically for particular subclasses and highlights some of the subclasses that lack an existing solution method. Our proposed unified framework for solving Procrustes problems is a significant contribution, as it covers all subclasses, including the challenging ones, where existing methods are limited in their applicability.

We demonstrated the correctness of our results in solving balanced orthogonal Procrustes problems of the form (85) for which the explicit solution (89) is known, as outlined in Subsection 5.1.3.1 and 5.1.3.2. To demonstrate the applicability of our proposed approach, we performed feature extraction for the Yale data set. As shown in Table 16, our proposed approach was slower than the OLSR algorithm [107]. However, as mentioned in [107], the l_1 norm is more appropriate for this application, as it is robust to outliers. Therefore, the proposed approach is a valuable alternative for solving unbalanced Procrustes problems with the l_1 norm, where the only existing method is a time-consuming differential approach [97]. Table 17 summarizes the comparison of the conic optimization approach with two existing methods. Although the proposed conic approach may not be as effective as existing methods tailored for specific subclasses of problems, it offers the advantage of applicability to a wider range of Procrustes problems. For example, it can handle weighted orthogonal Procrustes problems with the l_1, l_2, or l_∞ norm in the objective (see Tables 19, 20, and 21), weighted oblique Procrustes problems (see Table 27), and orthogonal Procrustes problems with additional linear constraints (see Tables 22 and 23). Finally, we applied the conic approach to solve the graph isomorphism problem, which we formulated as a two-sided orthogonal Procrustes problem of the form (99) and reformulated as a rank-constrained SDP problem (100). After solving this problem for several known pairs of isomorphic graphs, we were able to identify the corresponding permutation matrix in all cases, as demonstrated in Table 24. This discovery highlights the applicability of our proposed conic approach as a valuable alternative to existing methods, particularly when dealing with specific subclasses of Procrustes problems.

To conclude, the proposed conic optimization approach offers significant benefits in solving Procrustes problems with the l_1, l_2, and l_∞ norm and problems with additional linear or semidefinite constraints, where there are none or only slow solution methods. This approach provides a unified framework for matrix approximation problems in general, which may lead to a new perspective for analyzing these problems. Furthermore, the proposed conic approach could be extended to solve the so-called regularized minimization problems with the objective $\sum_{i=1}^{n} \|\mathcal{L}_i(X)\|$, where $\mathcal{L}_i$ are linear mappings and the matrix norms can be arbitrary, such as in [61] and [59]. On the other hand, the designed bisection algorithm (Algorithm 4) for solving rank-constrained optimization problems could

also inspire the development of algorithms to solve rank-constrained feasibility problems (36) that would converge to a low-rank solution. Besides applications addressed in this thesis, the proposed conic approach can be extended to solve other real-life problems, such as Euclidean distance matrix completion problems arising in wireless sensor networks or multidimensional scaling [29]. Overall, the proposed conic approach and bisection algorithm offer promising avenues for further research in the field of matrix approximation and optimization.

References

[1] S. Ahmed and I. M. Jaimoukha. A relaxation-based approach for the orthogonal procrustes problem with data uncertainties. In *Proceedings of 2012 UKACC International Conference on Control*, pages 906–911. IEEE, 2012.

[2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5, 1998.

[3] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

[4] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 8(3):746–768, 1998.

[5] M. S. Andersen, J. Dahl, L. Vandenberghe, et al. Cvxopt: A python package for convex optimization. *Available at cvxopt.org*, 54, 2013.

[6] L.-E. Andersson and T. Elfving. A constrained procrustes problem. *SIAM Journal on Matrix Analysis and Applications*, 18(1):124–139, 1997.

[7] M. Anjos, N. Higham, P. Takouda, and H. Wolkowicz. A semidefinite programming approach for the nearest correlation matrix problem. *University of Waterloo, Waterloo, Ontario, Canada, Preliminary Research Report*, 2003.

[8] M. ApS. Mosek modeling cookbook, 2020.

[9] K. Axiotis and M. Sviridenko. Local search algorithms for rank-constrained convex optimization. *arXiv preprint arXiv:2101.06262*, 2021.

[10] T. Bell. Global positioning system-based attitude determination and the orthogonal procrustes problem. *Journal of Guidance, Control, and Dynamics*, 26(5):820–822, 2003.

[11] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

[12] J. Berge. The rigid orthogonal procrustes rotation problem. *Psychometrika*, 71:201–205, 2006.

[13] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. Mixed-projection conic optimization: A new paradigm for modeling rank constraints. *Operations Research*, 70(6):3321–3344, 2022.

[14] C. Bogani, M. G. Gasparo, and A. Papini. A gss method for oblique l_1 procrustes problems. In *Applied And Industrial Mathematics In Italy III*, pages 87–98. World Scientific, 2010.

[15] A. Bojanczyk and A. Lutoborski. The procrustes problem for orthogonal stiefel matrices. *SIAM Journal on Scientific Computing*, 21, 2001.

[16] R. Borsdorf and N. J. Higham. A preconditioned newton algorithm for the nearest correlation matrix. *IMA Journal of Numerical Analysis*, 30(1):94–107, 2010.

[17] R. Borsdorf, N. J. Higham, and M. Raydan. Computing a nearest correlation matrix with factor structure. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2603–2622, 2010.

[18] S. Boyd and L. V. Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. 1999.

[19] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[20] A. Brace, D. G̨atarek, and M. Musiela. The market model of interest rate dynamics. *Mathematical finance*, 7(2):127–155, 1997.

[21] J. E. Brock. Optimal matrices describing linear systems. *AIAA Journal*, 6(7):1292–1296, 1968.

[22] J. A. Cadzow. Signal enhancement - a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(1):49–62, 1988.

[23] M. CHU and N. Trendafilov. On a differential equation approach to the weighted orthogonal procrustes problem. *Statistics and Computing*, 8, 1998.

[24] M. T. Chu and N. T. Trendafilov. The orthogonally constrained regression revisited. *Journal of Computational and Graphical Statistics*, 10(4):746–771, 2001.

[25] M. A. Cox and T. F. Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

[26] I. I. Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.

[27] J. Dattorro. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing USA, 2011.

[28] Y. Ding, D. Ge, and H. Wolkowicz. On equivalence of semidefinite relaxations for quadratic matrix programming. *Mathematics of Operations Research*, 36(1):88–104, 2011.

[29] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli. Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

[30] I. L. Dryden and K. V. Mardia. *Statistical shape analysis: with applications in R*, volume 995. John Wiley & Sons, 2016.

[31] X. Duan, J. Bai, M. Zhang, and X. Zhang. On the generalized low rank approximation of the correlation matrices arising in the asset portfolio. *Linear Algebra and its Applications*, 461:1–17, 2014.

[32] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[33] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20, 1998.

[34] L. Eldén and H. Park. A procrustes problem on the stiefel manifold. *Numerische Mathematik*, 82(4):599–619, 1999.

[35] K. Fan. On a theorem of weyl concerning eigenvalues of linear transformations. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

[36] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[37] M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. volume 6, pages 4734 – 4739, 2001.

[38] M. Fazel, H. Hindi, and S. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. volume 3, pages 2156 – 2162, 2003.

[39] M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proceedings of the 2004 American control conference*, volume 4, pages 3273–3278. IEEE, 2004.

[40] M. Fiori and G. Sapiro. On spectral properties for graph matching and graph isomorphism problems. *Information and Inference: A Journal of the IMA*, 4(1):63–76, 2015.

[41] J. Francisco and T. Martini dos Santos. Spectral projected gradient method for the procrustes problem. *Trends in Applied and Computational Mathmatics*, 15:83–96, 2014.

[42] J. B. Francisco and F. S. V. Bazán. Nonmonotone algorithm for minimization on closed sets with applications to minimization on stiefel manifolds. *Journal of Computational and Applied Mathematics*, 236(10):2717–2727, 2012.

[43] A. Fu, B. Narasimhan, and S. Boyd. CVXR: An R package for disciplined convex optimization. *Journal of Statistical Software*, 94(14):1–34, 2020.

[44] T. Fulová. Finding low-rank solutions in financial factor models. *Proceedings of the Conference Algoritmy*, pages 161–170, 2020.

[45] T. Fulová. Searching for low-rank solutions to semidefinite problems with a special structure. In *MMEI 2021 Mathematical Methods in Economy and Industry : Book of Abstracts and Conference Programme*, pages 31–31. Slovenská akadémia vied, Bratislava, 2021.

[46] T. Fulová. A conic optimization approach for solving Procrustes problems with quadratic constraints. In *ODS2022: Book of abstracts*, pages 172–172. Universita degli Studi di Firenze, Florencia, 2022.

[47] T. Fulová and M. Trnovská. Solving constrained Procrustes problems: a conic optimization approach. *submitted as arXiv preprint*, 2023.

[48] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[49] C. F. Gauß. *Werke: Theoria motus corporum coelestium in sectionibus conicis solem ambientum*, volume 7. Perthes et Besser, 1809.

[50] N. Gillis and P. Sharma. A semi-analytical approach for the positive semidefinite procrustes problem. *Linear Algebra and its Applications*, 540:112–137, 2018.

[51] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[52] J. Gower. Multivariate analysis. ordination, multidimensional scaling and allied topics. *Handbook of Applicable Mathematics, VI.: Statistics (B)*, 1984.

[53] J. Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.

[54] J. C. Gower and G. B. Dijksterhuis. *Procrustes problems*, volume 30. OUP Oxford, 2004.

[55] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, pages 95–110. Springer-Verlag Limited, 2008.

[56] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, 2014.

[57] I. Grubišić and R. Pietersz. Efficient rank reduction of correlation matrices. *Linear Algebra and its Applications*, 422:629–653, 2005.

[58] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. `https://www.gurobi.com`, 2023.

[59] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, 43:44, 2007.

[60] N. Higham. Computing the nearest correlation matrix - a problem from finance. *IMA Journal of Numerical Analysis*, 22, 2002.

[61] G. Huang, S. Noschese, and L. Reichel. Regularization matrices determined by matrix nearness problems. *Linear Algebra and its Applications*, 502:41–57, 2016. Structured Matrices: Theory and Applications.

[62] P. Jingjing, W. Qingwen, P. Zhenyun, and C. Zhencheng. Solution of symmetric positive semidefinite procrustes problem. *The Electronic Journal of Linear Algebra*, 35:543–554, 2019.

[63] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

[64] E. Klerk, C. Roos, and T. Terlaky. *Infeasible-start semidefinite programming algorithms via self-dual embeddings*, pages 215–236. 1998.

[65] M. A. Koschat and D. F. Swayne. A weighted procrustes criterion. *Psychometrika*, 56(2):229–239, 1991.

[66] J.-B. Lasserre and M. Anjos. *Handbook of Semidefinite, Conic and Polynomial Optimization*, volume 166. 2011.

[67] A. M. Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes.* chez Firmin Didot, libraire pour lew mathematiques, la marine, 1806.

[68] A. Lemon, A. So, and Y. Ye. Low-rank semidefinite programming: Theory and applications. 2:1–156, 2016.

[69] Q. Li and H.-d. Qi. A sequential semismooth newton method for the nearest low-rank correlation matrix problem. *SIAM Journal on Optimization*, 21(4):1641–1666, 2011.

[70] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1):193–228, 1998.

[71] The MathWorks Inc. *MATLAB version: 9.6.0 (R2019a), Natick, Massachusetts, United States*, 2022.

[72] E. Million. The hadamard product. *Course Notes*, 3(6):1–7, 2007.

[73] A. Minabutdinov, I. Manaev, and M. Bouev. Finding the nearest covariance matrix: the foreign exchange market case. *Journal of Computational Finance*, 24(2), 2018.

[74] A. Mooijaart and J. J. Commandeur. A general solution of the weighted orthonormal procrustes problem. *Psychometrika*, 55(4):657–663, 1990.

[75] J. Moré and D. Sorenson. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4:553 – 572, 1983.

[76] S. A. Mulaik. *Foundations of factor analysis*. CRC press, 2009.

[77] S. Naldi. Solving rank-constrained semidefinite programs in exact arithmetic. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 357–364, 2016.

[78] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[79] P. Parrilo and S. Lall. Semidefinite programming relaxations and algebraic optimization in control. *European Journal of Control*, 9:307–321, 2003.

[80] J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 18(1):186–205, 2007.

[81] R. Pietersz 4 and P. J. Groenen. Rank reduction of correlation matrices by majorization. *Quantitative Finance*, 4(6):649–662, 2004.

[82] H. Qi and D. Sun. A quadratically convergent newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 28(2):360–385, 2006.

[83] Y. Qiu and A. Wang. Solving balanced procrustes problem with some constraints by eigenvalue decomposition. *J. Computational Applied Mathematics*, 233:2916–2924, 2010.

[84] F. Rendl. Semidefinite relaxations for integer programming. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 687–726, 2010.

[85] F. Rendl. Semidefinite relaxations for partitioning, assignment and ordering problems. *Annals of Operations Research*, 240:119–140, 2016.

[86] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. *Mathematische Annalen*, 63(4):433–476, 1907.

[87] P. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966.

[88] I. Söderkvist and P.-Å. Wedin. On condition numbers and algorithms for determining a rigid body movement. *BIT Numerical Mathematics*, 34(3):424–436, 1994.

[89] J. F. Sturm. Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

[90] T. Suffridge and T. Hayden. Approximation by a hermitian positive semidefinite toeplitz matrix. *SIAM journal on matrix analysis and applications*, 14(3):721–734, 1993.

[91] C. Sun and R. Dai. Rank-constrained optimization and its applications. *Automatica*, 82:128–136, 2017.

[92] R. Takapoui and S. Boyd. Linear programming heuristics for the graph isomorphism problem. *arXiv preprint arXiv:1611.00711*, 2016.

[93] K. Tanioka, Y. Furotani, and S. Hiwa. Thresholding approach for low-rank correlation matrix based on MM algorithm. *Entropy*, 24(5):579, 2022.

[94] M. J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[95] K. C. Toh, M. J. Todd, and R. H. Tütüncü. Sdpt3 — a matlab software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1-4):545–581, 1999.

[96] N. T. Trendafilov. A continuous-time approach to the oblique procrustes problem. *Behaviormetrika*, 26:167–181, 1999.

[97] N. T. Trendafilov. On the l_1 procrustes problem. *Future Generation Computer Systems*, 19(7):1177–1186, 2003.

[98] N. T. Trendafilov and G. Watson. The l_1 oblique procrustes problem. *Statistics and Computing*, 14(1):39–51, 2004.

[99] T. Viklands and P.-Å. Wedin. Algorithms for linear least squares problems on the stiefel manifold. 2006.

[100] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming. Theory, algorithms, and applications*. 2000.

[101] Y. Yajima. *Convex envelopes in optimization problems*, pages 343–344. Springer, Boston, 2001.

[102] Y. Ye. *Interior point algorithms: theory and analysis*. John Wiley & Sons, 2011.

[103] F. Zhang. *Matrix theory: basic results and techniques*. Springer, 2011.

[104] L. Zhang, W. H. Yang, C. Shen, and J. Ying. An eigenvalue-based method for the unbalanced procrustes problem. *SIAM Journal on Matrix Analysis and Applications*, 41:957–983, 2020.

[105] Z. Zhang and K. Du. Successive projection method for solving the unbalanced procrustes problem. *Science in China Series A*, 49(7):971–986, 2006.

[106] Z. Zhang and L. Wu. Optimal low-rank approximation to a correlation matrix. *Linear algebra and its applications*, 364:161–187, 2003.

[107] H. Zhao, Z. Wang, and F. Nie. Orthogonal least squares regression for feature extraction. *Neurocomputing*, 216:200–207, 2016.

Appendix

A Matrix theory

A.1 Positive semidefinite matrices

In the following, we present three equivalent definitions of symmetric positive semidefinite matrices.

Definition A.1 ([8, §6]). *A symmetric matrix $X \in \mathbb{S}^n$ is called positive semidefinite if*

$$z^T X z \geq 0, \ \ \forall z \in \mathbb{R}^n.$$

Definition A.2 ([8, §6]). *A symmetric matrix $X \in \mathbb{S}^n$ is called positive semidefinite if*

$$\lambda_i \geq 0, \ \ \forall i = 1, ..., n,$$

where $\lambda_1, ..., \lambda_n$ are eigenvalues of X.

Using the spectral factorization of X, we have

$$z^T X z = z^T (\sum_{i=1}^{n} \lambda_i q_i q_i^T) z = \sum_{i=1}^{n} \lambda_i (z^T q_i)^2, \ \ \forall z \in \mathbb{R}^n, \tag{111}$$

where $q_i \in \mathbb{R}^n$ are the orthogonal eigenvectors of X. From (111), we obtain the statement of Definition A.2.

Definition A.3 ([8, §6]). *A symmetric matrix $X \in \mathbb{S}^n$ is called positive semidefinite if it is a Grammian matrix $X = V^T V$ for some $V \in \mathbb{R}^{n \times n}$.*

Using the Grammian representation, we have

$$z^T A z = z^T V^T V z = \|Vz\|_2^2 \geq 0, \ \ \forall z \in \mathbb{R}^n. \tag{112}$$

From the spectral decomposition $X = Q \Lambda Q^T$, where $\Lambda = diag(\lambda_1, ..., \lambda_n)$ and $QQ^T = I_n$, we can take $V = \Lambda^{\frac{1}{2}} Q^T$ in Definition (A.3).

Lemma A.1 ([8, §6]). *Assume X is a symmetric positive semidefinite matrix of size n. Then it possesses these properties:*

- *The diagonal entries of X are .*
- *A block-diagonal matrix $X = diag(X_1, ..., X_p)$ is positive semidefinite if and only if each block X_i $(i = 1, ..., n)$ is positive semidefinite.*
- *A quadratic transformation of X given as $M = B^T XB$, where B is a regular matrix, is positive semidefinite if and only if X is positive semidefinite.*
- *Any principal submatrix of X is positive semidefinite.*
- *The inner product of positive semidefinite matrices is .*
- *The pseudo-inverse $X^\dagger$ of X is positive semidefinite.*

Lemma A.2 ([103, §6.1])**.** *If $A \preceq B$, then $tr(A) \leq tr(B)$.*

Proof. Let $A \preceq B$. From definition of Löwner ordering we have $B - A \succeq 0$. Since any positive semidefinite matrix has nonnegative diagonal elements, it holds $tr(B - A) \geq 0$, which implies $tr(A) \leq tr(B)$. □

Lemma A.3. *Let $X \in \mathbb{S}_+$ have eigenvalues $\lambda_1 \geq ... \geq \lambda_n \geq 0$ and $r = rank(X)$. Then there exists $V \in \mathbb{R}^{m \times r}$ such that $X = VV^T$.*

Proof. Let $X = Q\Lambda Q^T = (Q\Lambda^{\frac{1}{2}})(Q\Lambda^{\frac{1}{2}})^T$, where $QQ^T = I_n$ and $\Lambda = diag(\lambda_1, ..., \lambda_n)$, be the spectral decomposition of X. Since $rank(X) = r$, $\lambda_{r+1} = ... = \lambda_n = 0$. It implies $X = VV^T$, where V consists of the first r columns of the matrix $Q\Lambda^{\frac{1}{2}}$. □

A.2 Hadamard product

Definition A.4 ([103, §6.5])**.** *The Hadamard product of two matrices A and B of the size same $m \times n$ is defined to be the entrywise product*

$$A \circ B = \begin{pmatrix} a_{11}b_{11} & ... & a_{1n}b_{11} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & ... & a_{mn}b_{mn} \end{pmatrix}.$$

Lemma A.4 ([72])**.** *Suppose $\alpha \in \mathbb{C}$, and A, B and C are $m \times n$ matrices.*

Then $C \circ (A + B) = C \circ A + C \circ B$. Furthermore, $\alpha(A \circ B) = (\alpha A) \circ B = A \circ (\alpha B)$.

B Conic optimization

B.1 Duality in conic optimization

Definition B.1 ([19, §2.6.1]). *Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a cone. Then the dual cone of $\mathcal{K}$ is defined as the set*

$$\mathcal{K}^* = \{y \mid x^T y \geq 0,\ \forall x \in \mathcal{K}\}. \tag{113}$$

If $\mathcal{K} \subseteq \mathbb{R}^{n\times n}$ we can rewrite the definition of a dual cone as follows

$$\mathcal{K}^* = \{Y \mid tr(X^T Y) \geq 0,\ \forall X \in \mathcal{K}\}. \tag{114}$$

Geometrically, Definition B.1 states that any y belongs to the dual cone $\mathcal{K}^*$ if and only if $-y$ is the normal of a hyperplane that supports $\mathcal{K}$ at the origin ([19, §2.6]). Let us note that the dual cone $\mathcal{K}^*$ is always a convex cone, even if cone $\mathcal{K}$ is nonconvex.

The Lagrange dual of conic problem (11) can be written in the form ([27, §4.1])

$$\begin{array}{lrcl} \underset{y,s}{max} & b^T y & & \\ & A^T y + s & = & c, \\ & s & \in & \mathcal{K}^*, \end{array} \tag{115}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ are the given data, and $\mathcal{K}^*$ is the dual cone of $\mathcal{K}$.

If the problem variable is a matrix $X \in \mathbb{R}^{n\times n}$, the conic problem (11) has the form

$$\begin{array}{lrcl} \underset{X\in\mathbb{R}^{n\times n}}{min} & tr(C^T X) & & \\ & tr(A_i X) & = & b_i,\ i = 1, ..., m, \\ & X & \in & \mathcal{K}, \end{array} \tag{116}$$

where $C, A_1, ..., A_m \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^m$ and $\mathcal{K}$ is a convex cone.

Consequently, its dual is formulated in the form

$$\begin{array}{lrcl} \underset{y\in\mathbb{R}^m, S\in\mathcal{K}}{max} & b^T y & & \\ & \sum_{i=1}^{m} y_i A_i + S & = & C, \\ & S & \in & \mathcal{K}^*, \end{array} \tag{117}$$

where $C, A_1, ..., A_m \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^m$ are the given data and $\mathcal{K}^*$ is the dual cone of $\mathcal{K}$.

The orthant, the positive semidefinite cone, and the second-order cone are self-dual cones, i.e., $\mathcal{K} = \mathcal{K}^*$ (see [19, §2.6]). This property ensures optimization over the same cone in primal and dual problems, enabling the application of the same techniques to solve both. However, it does not hold, in general. For example, if we take the l_1-norm cone, its dual is the l_∞-norm cone, and the dual cone of the copositive cone is the cone of completely positive matrices. In such cases, both cones must be analyzed.

B.2 Eigenvalue optimization

Proposition B.1 ([8, §6.2]). *Let X be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Then the largest eigenvalue can be characterized in epigraph form $\lambda_1 \leq t$ as $tI_n - X \succeq 0$.*

Proof. Let $X = Q\Lambda Q^T$ be a spectral decomposition of X where $Q^TQ = I_n$ and $\Lambda = diag(\lambda_1, ..., \lambda_n)$. Then $\lambda_1 \leq t$ if and only if $Q^T(tI_n - X)Q = tI_n - \Lambda \succeq 0$. □

Proposition B.2 ([8, §6.2]). *Let X be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Then the smallest eigenvalue can be characterized in hypograph form $\lambda_n \geq t$ as $X - tI_n \succeq 0$.*

Proof. Let $X = Q\Lambda Q^T$ be a spectral decomposition of X where $Q^TQ = I_n$ and $\Lambda = diag(\lambda_1, ..., \lambda_n)$. Then $\lambda_n \geq t$ if and only if $Q^T(tI_n - X)Q = tI_n - \Lambda \preceq 0$. □

Analogical approaches can be used to model the eigenvalue spread $\lambda_1 - \lambda_m$ and the spectral radius $\rho(X) = \max_i |\lambda_i|$. Below, we present characterizations of the sum of the largest k eigenvalues and the sum of the smallest $n-k$ eigenvalues of X. Before proceeding, we recall several auxiliary statements.

Lemma B.1 ([35]). *Let $\lambda_1 \geq ... \geq \lambda_k \geq ... \geq \lambda_n$ be eigenvalues of matrix $X \in \mathbb{S}^n$. Then*

a)

$$\sum_{i=1}^{k} \lambda_i = sup\{tr(V^TXV) \mid V \in \mathbb{R}^{n\times k}, V^TV = I_k\}. \tag{118}$$

b)

$$\sum_{i=k+1}^{n} \lambda_i = inf\{tr(V^TXV) \mid V \in \mathbb{R}^{n\times (n-k)}, V^TV = I_{n-k}\}. \tag{119}$$

Proposition B.3 ([2, §4.1])**.** *Let X be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Then the sum of its k largest eigenvalues can be characterized as the optimal value of the semidefinite program*

$$\begin{aligned} \lambda_1 + ... + \lambda_k \quad = \underset{U \in \mathbb{S}^n}{max} \quad & tr(XU) \\ & tr(U) = k, \\ & 0 \preceq U \preceq I_n. \end{aligned} \tag{120}$$

Proof. Lemma B.1a) enables to express the sum of k largest eigenvalues as a supremum (118). If we set $U = \tilde{V}\tilde{V}^T$, where $0 \preceq \tilde{V}\tilde{V}^T \preceq I_n$ and $tr(\tilde{V}\tilde{V}^T) = k$, we obtain the problem in the form (120). □

Proposition B.3 offers a way to find the sum of the k largest eigenvalues of a symmetric matrix as a solution of a semidefinite program. Similarly, we can also express the sum of the $n - k$ smallest eigenvalues, as stated in Proposition B.4.

Proposition B.4. *Let X be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Then the sum of its $n - k$ smallest eigenvalues can be characterized as the optimal value of the semidefinite program*

$$\begin{aligned} \lambda_{k+1} + ... + \lambda_n \quad = \underset{U \in \mathbb{S}^n}{min} \quad & tr(XU) \\ & tr(U) = n - k, \\ & 0 \preceq U \preceq I_n. \end{aligned} \tag{121}$$

Proof. Lemma B.1b) enables to express the sum of $n - k$ smallest eigenvalues as an infimum (119). If we set $U = \tilde{V}\tilde{V}^T$, where $0 \preceq \tilde{V}\tilde{V}^T \preceq I_n$ and $tr(\tilde{V}\tilde{V}^T) = n - k$, we obtain the problem in the form (121). □

B.3 Transformations of norm minimization problems

The l_1 norm

Consider the problem (21) with the l_1 norm in the objective, that is,

$$\begin{aligned} \min \quad & f(X) := \|\mathcal{L}(X)\|_1 \\ & X \in \mathcal{P}. \end{aligned} \tag{22}$$

Using the definition of the l_1 norm, also known as the max-column-sum norm, we can rewrite (22) into the form

$$\begin{array}{ll} \min & \max\limits_{1\leq j\leq q} \sum_{i=1}^{p} \mid \mathcal{L}(X)_{ij} \mid \\ & X \in \mathcal{P}. \end{array} \tag{122}$$

Introducing a new variable $t \in \mathbb{R}$, we obtain an equivalent problem to (122):

$$\begin{array}{lll} \min & t & \\ & X & \in \mathcal{P}, \\ & \sum_{i=1}^{p} \mid \mathcal{L}(X)_{ij} \mid & \leq t. \end{array} \tag{123}$$

To deal with the sum of absolute values in the last constraint of (123), we introduce a new matrix variable $S \in \mathbb{R}^{p\times q}$ and obtain the reformulation of (123) in the form

$$\begin{array}{lll} \min & t & \\ & X & \in \mathcal{P}, \\ -S_{ij} & \leq \mathcal{L}(X)_{ij} & \leq S_{ij}, \quad \forall i = 1, ..., p, \ \forall j = 1, ..., q, \\ & \sum_{i=1}^{p} S_{ij} & \leq t \qquad \forall j = 1, ..., q. \end{array} \tag{124}$$

Assuming that the sign "$\leq$" stands for a component-wise inequality, a compact notation of (124) can be formulated as

$$\begin{array}{lll} \min & t & \\ & X & \in \mathcal{P}, \\ -S & \leq \mathcal{L}(X) & \leq S, \\ & S^T \mathbf{1}_p & \leq t\mathbf{1}_q. \end{array} \tag{23}$$

The derived result is summarized in Lemma 2.2.

The spectral norm

Consider the problem (21) with the spectral norm in the objective:

$$\begin{array}{ll} \min & f(X) := \|\mathcal{L}(X)\|_2 \\ & X \in \mathcal{P}. \end{array} \tag{26}$$

If we define the variable $s \in \mathbb{R}$ and apply the epigraph transformation, the problem (26) can be equivalently rewritten as

$$\begin{array}{ll} \min & s \\ & X \in \mathcal{P} \\ & \|\mathcal{L}(X)\|_2 \leq s. \end{array} \tag{125}$$

After using the definition of the spectral norm $\|Y\|_2 = \sigma_{max}(Y)$ where σ_{max} stands for the largest singular value of Y, the last constraint of (125) can be expressed as

$$\sigma_{max}\Big(\mathcal{L}(X)\Big) \leq s.$$

Consequently, we rewrite this constraint as

$$\lambda_{max}(\mathcal{L}(X)^T\mathcal{L}(X)) \leq s^2,$$

$$\lambda_{min}(s^2 I_q - \mathcal{L}(X)^T\mathcal{L}(X)) \geq 0,$$

where λ_{max} denotes the largest eigenvalue and λ_{min} the smallest eigenvalue of the particular matrix. It follows that problem (125) can be rewritten as the problem

$$\begin{aligned} \min \quad & s \\ & X \in \mathcal{P} \\ & s^2 I - \mathcal{L}(X)^T\mathcal{L}(X) \succeq 0. \end{aligned} \tag{27}$$

The result is presented in Lemma 2.4.

The Frobenius norm

Consider the problem (21) with the Frobenius norm in the objective:

$$\begin{aligned} \min \quad & f(X) := \|\mathcal{L}(X)\|_F \\ & X \in \mathcal{P}. \end{aligned} \tag{28}$$

Using Frobenius norm definition $\|Y\|_2 = \sqrt{tr(YY^T)}$ and monotone increasing transformation, the objective of (28) can be rewritten as

$$\|\mathcal{L}(X)\|_F^2 = tr\Big(\mathcal{L}(X)\mathcal{L}(X)^T\Big).$$

After introducing a new variable $Z \in \mathbb{R}^{p\times p}$ such that

$$Z \succeq \mathcal{L}(X)\mathcal{L}(X)^T,$$

we can apply Lemma A.2 to obtain an upper bound on the squared objective

$$tr(Z) \geq tr\Big(\mathcal{L}(X)\mathcal{L}(X)^T\Big) = \|\mathcal{L}(X)\|_F^2.$$

As a result, the problem (28) can be reformulated as

$$\begin{aligned} \min \quad & tr(Z) \\ & X \in \mathcal{P} \\ & Z \succeq \mathcal{L}(X)\mathcal{L}(X)^T. \end{aligned} \tag{29}$$

This derived result is summarized in Lemma 2.5.

B.4 Representations of nonconvex quadratic constraints

Lemma B.2. *Let $X \in \mathbb{R}^{m\times n}$, $G \in \mathbb{S}^n$ and $Y \in \mathbb{S}^n$. Then the following constraints are equivalent:*

i) $X^T X \succeq G$,

ii) $Y - G \succeq 0 \ \wedge \ X^T X = Y$.

Proof. Let $X^T X \succeq G$ and define $Y = X^T X$. It follows that $Y \succeq G$. Reversely, if $Y \succeq G$ and $X^T X = Y$ then $X^T X \succeq G$. □

Note that if Lemma 2.1 is applied to the nonconvex quadratic representation $X^T X = Y$ in the statement ii) of Lemma B.2, we obtain the representation of the nonconvex quadratic constraint $X^T X \succeq G$ from Table 2.

Lemma B.3. *Let $X \in \mathbb{R}^{m\times n}$, $G \in \mathbb{S}^n$, and $g \in \mathbb{R}$. Then the following constraints are equivalent:*

i) $tr(X^T X) \leq g$,

ii) $G \succeq X^T X \ \wedge \ tr(G) \leq g$.

Proof. Let $tr(X^T X) \leq g$ and define $G = X^T X$. Then, $tr(G) = tr(X^T X)$ implying $tr(G) \leq g$. Reversely, if $tr(G) \leq g$ and $G \succeq X^T X$, then, according to Lemma A.2, $tr(X^T X) \leq tr(G) \leq g$. □

Lemma B.4. *Let $X \in \mathbb{R}^{m\times n}$, $G \in \mathbb{S}^n$, and $g \in \mathbb{R}$. Then the following constraints are equivalent:*

i) $tr(X^T X) \geq g$,

ii) $G = X^T X \ \wedge \ tr(G) \leq g$.

Proof. Let $tr(X^T X) \geq g$ and define $G = X^T X$. Then, $tr(G) = tr(X^T X)$ implying $tr(G) \geq g$. Reversely, if $tr(G) \geq g$ and $G = X^T X$, then $tr(X^T X) \geq g$ according to Lemma A.2. □

If Lemma 2.1 is applied to the nonconvex quadratic representation $tr(X^TX) \leq g$ in the statement ii) of Lemma B.3 and on $tr(X^TX) \geq g$ from ii) of Lemma B.4, we obtain the representations of the nonconvex quadratic constraints $tr(X^TX) \leq g$ and $tr(X^TX) \geq g$ from Table 2. Analogically, we can obtain a representation of $tr(X^TX) = g$.

Lemma B.5. *Let $X \in \mathbb{R}^{m\times n}$, $G \in \mathbb{S}^n$, and $h \in \mathbb{R}^n$. Then the following constraints are equivalent:*

i) $diag(X^TX) \leq h$,

ii) $G = X^TX \ \wedge \ diag(G) \leq h$.

Proof. Let $diag(X^TX) \leq h$ and define $G = X^TX$. Then, $diag(G) = diag(X^TX)$ implying $diag(G) \leq h$. Reversely, if $diag(G) \leq h$ and $G = X^TX$, then $diag(X^TX) \leq \ h$. □

If Lemma 2.1 is applied to the nonconvex quadratic representation $diag(X^TX) \leq h$ in the statement ii) of Lemma B.5, we obtain the representation of the nonconvex quadratic constraint $diag(X^TX) \leq h$ from Table 2. Analogically, we can obtain representations of $diag(X^TX) = h$ and $diag(X^TX) \geq h$.

C Convex envelope of the rank function

C.1 Definition of the convex envelope

Definition C.1 ([101]). *Let $f : \mathcal{C} \to \mathbb{R}$, where $\mathcal{C} \subseteq \mathbb{R}^n$. The function cenv f is a convex envelope of the function f on the set $\mathcal{C}$, if*

i) cenv f is a convex function,

ii) cenv $f(x) \leq f(x), \forall x \in \mathcal{C}$,

iii) for any convex function $\varphi(x)$ such that $\varphi(x) \leq f(x)\ \forall x$ holds $\varphi(x) \leq$ cenv $f(x)\ \forall x$.

In other words, the convex envelope of the function f is such a convex function *cenv* f, which is the pointwise closest to the function f among all convex functions minorizing f. An illustration of the convex envelope of the function f is shown in Figure 16. We can also see that the epigraph of the convex envelope is the smallest convex set that includes the epigraph of the given function f ([19, §3]).

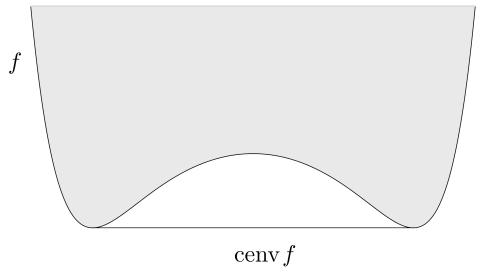


Figure 16: Illustration of the convex envelope of the function f.

C.2 Trace as the convex envelope of the rank function

Theorem C.1 ([37]). *The convex envelope of the function rank(X) on the set*

$$\mathcal{U}_1 = \{X \in \mathbb{S}^n_+ \mid 0 \preceq X \preceq I_n\} \tag{126}$$

is the function tr(X), that is,

$$\textit{cenv}\ \mathrm{rank(X)} = \mathrm{tr(X)}. \tag{127}$$

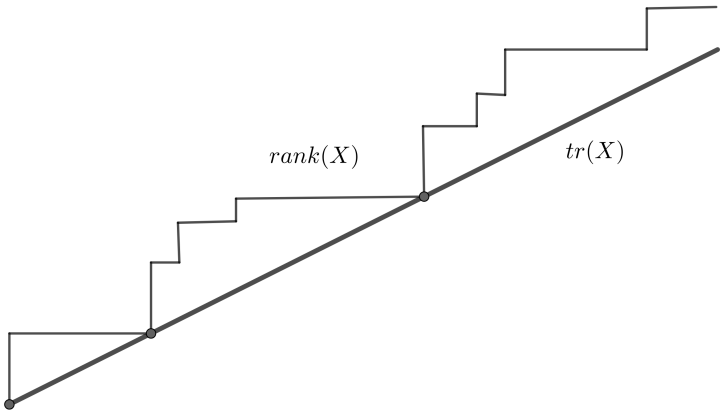


Figure 17: Illustration of the matrix trace as the convex envelope of the rank. Intersection points denote projection matrices, for which we have $rank(X) = tr(X)$.

The proof of Theorem C.1 was introduced in [37] for a general $X \in \mathbb{R}^{m \times n}$ and is based on the known convex optimization result that the conjugate of the conjugate function is the convex envelope of the given function f, i.e., $f^{**} = \text{cenv f}$ ([37]). Regarding only the case of symmetric positive semidefinite matrices, we propose this proof only using the convex envelope definition (Definition C.1). We recommend looking at Figure 17 to understand the proof better.

Proof. We want to show that the trace function satisfies Definition C.1 for $f(X) = rank(X)$ and $\mathcal{C} = \mathcal{U}_1$. Property i) is satisfied since the trace is linear in X. The rank of a semidefinite matrix is the number of its nonzero eigenvalues, and the condition $0 \preceq X \preceq I_n$ ensures that $\lambda_i \in [0, 1]\ \forall i$. Therefore, if $rank(X) = k$, then $tr(X) \leq k$ and property ii) is valid. It remains to show the property iii), that is, we want to prove that if a convex function $\varphi(X) \leq rank(X)$ then $\varphi(X) \leq tr(X)\ \forall X \in \mathcal{U}_1$. Let us proceed with mathematical induction on the rank of the matrix. We introduce the notation X^k for a rank-k matrix.

For $k = 0$, the statement is apparently true since only $X = 0$ comes into consideration.

For $k = 1$, a rank-1 matrix can be expressed as

$$X^1 = vv^T, \text{ for some vector } v \neq 0. \tag{128}$$

The only nonzero eigenvalue of X^1 is then $\lambda = v^T v$. Since $X^1 \in \mathcal{U}_1$, $\lambda \in [0, 1]$. Moreover, we can construct a projection matrix $P = \frac{vv^T}{v^T v}$ satisfying $P \succeq 0$, $P = P^T$, $P = P^2$ and

having eigenvalues 0 or 1. Using this, we can rewrite X^1 as a convex combination of the projection matrix P and the zero matrices as follows

$$X^1 = \lambda \frac{vv^T}{\lambda} + (1-\lambda)0 = \lambda P + (1-\lambda)X^0.$$

Then, for a convex function φ, we have

$$\varphi(X^1) \leq \lambda\varphi(P) + (1-\lambda)\varphi(X^0) \leq \lambda rank(P) + (1-\lambda)rank(X^0),$$

where the second inequality comes from the assumption $\varphi(X^1) \leq rank(X^1)$. Note that any projection matrix has the property $rank(P) = tr(P)$. Consequently, we obtain

$$\varphi(X^1) \leq \lambda rank(P) + (1-\lambda)rank(X^0) = \lambda tr(P) + (1-\lambda)tr(X^0) = tr(X^1).$$

Let us formulate the induction assumption: Let the statement be true for matrices with a rank lower than m, i.e.

$$\varphi(X^k) \leq rank(X^k) \Rightarrow \varphi(X^k) \leq tr(X^k),\ \forall k = 0, ..., m-1. \tag{129}$$

We will show that this statement also holds for rank-m matrices.

Let $\lambda_1 \geq ... \geq \lambda_m$ be nonzero eigenvalues of matrix X^m and $q_1, ..., q_n$ its corresponding orthogonal eigenvectors. Then

$$X^m = \sum_{i=1}^{m} \lambda_i q_i q_i^T = \lambda_1 q_1 q_1^T + \sum_{i=2}^{m} \lambda_i q_i q_i^T. \tag{130}$$

Since $0 \preceq X^m \preceq I_n$, then $\lambda_1 \in (0, 1]$ and the matrix X^m can be expressed as a convex combination of a rank-1 matrix and a rank-$(m-1)$ matrix as follows

$$X^m = \lambda_1 q_1 q_1^T + (1-\lambda_1)\sum_{i=2}^{m} \frac{\lambda_i}{1-\lambda_1} q_i q_i^T = \lambda_1 X^1 + (1-\lambda_1)X^{m-1}. \tag{131}$$

Note that $X^{m-1} \coloneqq \sum_{i=2}^{m} \frac{\lambda_i}{1-\lambda_1} q_i q_i^T$ is still a matrix with rank equal to $m-1$ because multiplication by a positive constant does not affect the matrix rank.

From the convexity of function φ and the induction assumption (129), we have

$$\begin{aligned} \varphi(X^m) &= \varphi(\lambda_1 X^1 + (1-\lambda_1)X^{m-1}) \leq \lambda_1\varphi(X^1) + (1-\lambda_1)\varphi(X^{m-1}) \\ &\leq \lambda_1 tr(X^1) + (1-\lambda_1)tr(X^{m-1}) = tr(X^k). \end{aligned} \tag{132}$$

□

Proposition C.1 ([36]). *The convex envelope of the function rank(X) on the set*

$$\mathcal{U}_\mu = \{X \in \mathbb{S}_+^n \mid 0 \preceq X \preceq \mu I_n, \mu \in \mathbb{N}\} \tag{133}$$

is the function $\frac{1}{\mu}tr(X)$*, i.e.*

$$cenv\,\mathrm{rank(X)} = \frac{1}{\mu}\mathrm{tr(X)}. \tag{134}$$

Proof. Let $X \in \mathcal{U}_\mu$. Set $Y = \frac{1}{\mu}X$. Then $Y \in \mathcal{U}_1$ and according to Theorem C.1 it holds

$$\mathrm{cenv\,rank(Y)} = \mathrm{tr(Y)} = \mathrm{tr}\left(\frac{1}{\mu}\mathrm{X}\right) = \frac{1}{\mu}\mathrm{tr(X)}. \tag{135}$$

Furthermore, since multiplication by a positive constant does not change the rank of the matrix, we have $rank(X) = rank(Y)$, and also cenv rank(X) = cenv rank(Y). □

Proposition C.1 obviously implies that for $X \in \mathcal{U}_\mu$ we have

$$tr(X) \leq \mu\, rank(X). \tag{136}$$

To confirm this result, it is sufficient to realize that the eigenvalues of matrix $X \in \mathcal{U}_\mu$ satisfy

$$\mu \geq \lambda_1 \geq ... \geq \lambda_n \geq 0. \tag{137}$$

Furthermore, the rank is equal to the number of nonzero eigenvalues, and if we handle a rank-k matrix, we clearly get

$$\mu \geq \lambda_1 \geq ... \geq \lambda_k > 0 = \lambda_{k+1} = ... = \lambda_n. \tag{138}$$

The relation (136) could be rewritten into the form

$$\lambda_1 + ... + \lambda_k \leq \mu\, k. \tag{139}$$

Obviously, the given statement holds because every nonzero eigenvalue is bounded by its highest possible value from above.

D Rank minimization heuristics

D.1 Rank minimization problem with a general matrix variable

In this section, we present adjustments to be applied while minimizing the rank of a general matrix such that we could use the trace heuristic. In the first step, the following theorem is applied. The proof can be found in [38].

Theorem D.1 ([38]). *Let $X \in \mathbb{R}^{m\times n}$ be a given matrix. Then $rank(X) \leq r$ if and only if there exist symmetric matrices $Y \in \mathbb{R}^{m\times m}$ and $Z \in \mathbb{R}^{n\times n}$ such that*

$$rank(Y) + rank(Z) \leq 2r \quad and \quad \begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0.$$

Using Theorem D.1, the rank minimization problem (38) can be converted into the form

$$\begin{aligned} \min_{X,Y,Z} \quad & \frac{1}{2}(rank(Y) + rank(Z)) \\ & \begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0, \\ & X \in \mathcal{C}. \end{aligned} \tag{140}$$

Although the problem (140) still contains the nonconvex rank function in its objective, we can apply the trace heuristic to solve it after rewriting the objective function as follows

$$rank(Y) + rank(Z) = rank\left(\begin{bmatrix} Y & 0 \\ 0 & Z \end{bmatrix}\right).$$

Since matrix $\begin{bmatrix} Y & 0 \\ 0 & Z \end{bmatrix}$ is symmetric and positive semidefinite, we are allowed to apply the trace heuristic here, and we obtain the objective function of the form

$$\frac{1}{2}tr\left(\begin{bmatrix} Y & 0 \\ 0 & Z \end{bmatrix}\right) = \frac{1}{2}(tr(Y) + tr(Z)).$$

Finally, we solve the following semidefinite problem

$$\begin{aligned} \underset{X,Y,Z}{min} \quad & \frac{1}{2}(tr(Y)+tr(Z)) \\ & \begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0, \\ & X \in \mathcal{C}. \end{aligned} \tag{141}$$

D.2 Concavity of the log-det function

Lemma D.1 ([19, §3.1])**.** *Function $f(X) = \log\det(X)$ is concave in X.*

Proof. Let us define

$$g(t) = f(Z + tV), \;\; t \in \mathbb{R}, \;\; V, Z \in \mathbb{S}^n. \tag{142}$$

We want to show that $g(t)$ is concave in t. We have

$$g(t) = \log\det(Z+tV) = \log\det\left(Z^{\frac{1}{2}}(I_n + tZ^{-\frac{1}{2}}VZ^{-\frac{1}{2}})Z^{\frac{1}{2}}\right) \tag{143}$$

$$= \log\det\left((I_n + tZ^{-\frac{1}{2}}VZ^{-\frac{1}{2}})Z\right) \tag{144}$$

$$= \log\left(\det(I_n + tZ^{-\frac{1}{2}}VZ^{-\frac{1}{2}})\det Z\right) \tag{145}$$

$$= \log\det(I_n + tZ^{-\frac{1}{2}}VZ^{-\frac{1}{2}}) + \log\det Z \tag{146}$$

After denoting the i-th eigenvalue of matrix $Z^{-\frac{1}{2}}VZ^{-\frac{1}{2}}$ by λ_i, we can write

$$g(t) = \log\prod_i (1+t\lambda_i) + \log\det Z \tag{147}$$

$$= \sum_i \log(1+t\lambda_i) + \log\det Z. \tag{148}$$

The first derivative of the function g with respect to t is

$$g'(t) = \sum_i \frac{\lambda_i}{1+t\lambda_i} \tag{149}$$

and its second derivative with respect to t has the form

$$g''(t) = -\sum_i \frac{\lambda_i^2}{(1+t\lambda_i)^2}. \tag{150}$$

Since $g''(t) \leq 0$, then g is concave in t and f is concave in X. □

D.3 Local minimization of the log-det function

The first Taylor series of the function $\log\det(X+\delta I_n)$ about X_k is given by ([38])

$$\log\det(X+\delta I_n) \approx \log\det(X_k+\delta I_n) + tr(X_k+\delta I_n)^{-1}(X-X_k), \tag{151}$$

where the gradient of the function $\log\det$ is

$$\bigtriangledown \log\det(X_k+\delta I_n) = (X_k+\delta I_n)^{-1}, \tag{152}$$

since

$$\frac{\partial}{\partial X_{ij}}\log\det(X_k+\delta I_n) = \frac{1}{det(X_k+\delta I_n)}\frac{\partial det(X_k+\delta I_n)}{\partial X_{ij}} \tag{153}$$

$$= \frac{1}{det(X_k+\delta I_n)} adj(X_k+\delta I_n)_{ji} \tag{154}$$

$$= (X_k+\delta I_n)^{-1}_{ji}. \tag{155}$$

Therefore, the minimum of the function $\log\det(X+\delta I_n)$ over the feasible set $\mathcal{C}$ can be found by iterative minimization of its local linearization (151). After neglecting constants, we obtain the iterative method from Section 3.2.2:

$$X_{k+1} = \underset{X\in\mathcal{C}}{\operatorname{argmin}}\, tr((X_k+\delta I_n)^{-1}X). \tag{43}$$