Semidefinite Relaxation Approaches for the Quadratic Assignment Problem

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte Dissertation

von

Marko Lange

aus Rostock

2016

Gutachter: **Prof. Dr. Siegfried M. Rump**
Institut für Zuverlässiges Rechnen, Technische Universität Hamburg

Prof. Dr. Dr.h.c. Frerich Keil
Institut für chemische Reaktionstechnik, Technische Universität Hamburg

Prof. Dr. Franz Rendl
Institut für Mathematik, Universität Klagenfurt

Datum der mündliche Prüfung: **13. Juni 2016**

Abstract

Nowadays, the quadratic assignment problem (QAP) is widely considered as one of the hardest of the NP-hard problems. The main reason why it is considered as such can be found in the enormous difficulty of computing good quality bounds when applying a branch-and-bound algorithm. The practice shows that - even with the use of today's computers - QAPs of size $n > 30$ are typically recognized as huge and hardly tractable computational problems.

In this work, we are concerned with the design of new semidefinite programming relaxations for the computation of lower bounds of the QAP. Various ways to improve the bounding programs upon their semidefinite relaxation bases are discussed and different reformulation procedures for the construction of efficiently solvable programming problems are presented. For a further tightening of the discussed relaxation programs, we exploit different cutting strategies and compile stronger versions of the corresponding cuts. Another key area of this thesis is the design of numerically stable implementations which are suitable for the computation of rigorous bounds.

The thesis is concluded with a large number of numerical examples to demonstrate the applicability of our methods.

Acknowledgements

This thesis evolved over a period of about four years. During this time a lot of people influenced this thesis directly and indirectly. Among the many people who deserve thanks, some are particularly prominent.

First and foremost, my appreciation goes to my advisors Professor Siegfried M. Rump and Priv.-Doz. Christian Jansson. Without their guidance and advice, this thesis would not have been possible. I wish to thank Professor Rump for the hours of fruitful and less fruitful discussions on various topics, for providing me with a wonderful work environment, for overlooking the creative chaos on and around my desktop, and for always having a good idea whenever a new issue occurred. He is undeniable the person who is liable for awakening my interest in computations that are reliable. No less I wish to thank Christian Jansson. He is the one who started my interest in disciplined convex programming and combinatorial optimization. His in-depth knowledge about optimization and his cheerful attitude helped me when I felt like hitting an impasse. There is just no way to thank him sufficiently for spending so many hours together with me bringing this thesis into a legible form.

I also would like to thank my colleagues, all staff of the Institute for Reliable Computing of the Hamburg University of Technology. Our technical staff Siegfried Kubon for always coming up with the right solution to some technical problem and for keeping the systems running. Horst Meyer for having a comment on every issue, for organizing all the finances, and for providing me with the right working tools. I could not have enjoyed my time at TUHH so much if it was not for Dirk, Florian, Kai, Malte, Ole, Prashant, and Robert. They provided me with the necessary distractions and were always available for discussions, especially these which were not related to research. In particular, I want to thank Kai and Prashant for helping me enhancing the legibility and comprehensibility of this thesis. Additionally, I would like to thank Kai for always having time and an open ear when I needed to tell somebody about my last "discoveries". Finally, I want to

acknowledge our secretary Ulrike Schneider for an always open office, for her upbeat and compassionate attitude, and for the wonderful stories during coffee breaks.

I am thankful to my friends from Hamburg, Elmshorn, Karlsruhe, and Rostock. Fan, Jan-Henning, Julijan, Katrin, Mara, Sebastian, Stephanie, Richard, Thorsten, and Yanti not only supported me with my decisions, they helped me to grow into the person I am, and always remind me of other important things in life.

Lastly, I would like to express how much I am indebted to my family. I am forever grateful to my father for his unwavering love, his strict education, and the incredible many hours he spent to help me with the basics in elementary school until it finally sunk in. No less gratitude goes to my mother for showing me the power of enthusiasms, for her deep faith and support, and for her strong love, even if it often shows up in embarrassing questions about my life. A special thank goes also to my younger brother. Thank you for being you, for letting me be me, and for being an upbeat roommate for the last two and a half years.

Contents

List of tables **ix**

1. Introduction **1**
1.1. The quadratic assignment problem 1
1.2. Notation and preliminary assumptions 3
1.3. Outline and contributions 6

2. Preparations **11**
2.1. Auxiliary results and further preparations 11

3. Semidefinite Programming Relaxations **23**
3.1. Relaxation concepts 23
3.1.1. The vector lifting approach 23
3.1.2. Matrix lifting 26
3.1.3. The matrix splitting approach 27
3.1.4. Relaxation of the eigenspace 29
3.2. Comparison and other relaxation properties 30

4. Reformulation Approaches **35**
4.1. Reduction via QAP reformulation 35
4.1.1. QAP reformulations 35
4.1.2. Non-redundant positive semidefinite matrix splitting 48
4.1.3. Commutation of data matrices 51
4.2. Reduced relaxation frameworks 52
4.3. Level-1 relaxations 55

5. New Relaxation Frameworks **59**
5.1. New relaxation approaches 59
5.1.1. Reduction via approximation 60
5.1.2. Matrix lifting revisited 65

5.1.3. Interrelated matrix splitting 68
5.2. Intermediate comparison of level-1 relaxations 77

6. Cutting Strategies **83**
6.1. Cuts 83
6.1.1. Gilmore-Lawler bound constraints 83
6.1.2. Eigenvalue related cuts 87
6.1.3. Linear bound constraints 97
6.1.4. Cuts based on p-norm conditions 101
6.2. Level-2 relaxations 105

7. Implementation and Numerical Results **109**
7.1. Motivation 109
7.2. Numerical difficulties 111
7.2.1. Ill-posed programming problems 113
7.2.2. Regularization 115
7.2.3. Minimal face representation 117
7.2.4. Remarks on applicability of regularizing procedures 121
7.2.5. Rigorous bounds via verification methods 128
7.3. Implementation details 133
7.3.1. Formulation in dual or primal standard form 134
7.3.2. Further implementation details and low-rank coefficient structures 139
7.4. Numerical results 143

8. Conclusion **155**
8.1. Future prospects 156

A. sdprQAP Quick Reference **159**

Constants & Sets **163**

Notation **165**

Bibliography **167**

List of tables

3.1. Comparison of basic relaxation concepts (level-0) 31

4.1. Comparison of reformulated relaxations (level-1) 58

5.1. Improvements related to new relaxation techniques 77

5.2. Bound computations for nonzero approximation threshold 78

7.1. Speed and accuracy improvements via reformulation 111

7.2. Rigorous bounds for ill-posed MS_1 instances 132

7.3. Rigorous bounds for well-posed MS_1 instances 133

7.4. Comparison of high-dimensional relaxations 144

7.5. Comparison of low-dimensional relaxations 147

7.6. New best known bounds 153

A.1. User functions in sdprQAP 160

A.2. Options for sdprQAP 161

Chapter 1.

Introduction

1.1. The quadratic assignment problem

The history of the *Quadratic Assignment Problem* (QAP) dates back to 1957, when Koopmans and Beckmann [59] introduced it as a mathematical model for problems in the allocation of indivisible resources. Its problem class entails a great number of applications from different scenarios in the topic of combinatorial optimization. This includes problems arising in location theory, facility layout, VLSI design, process communications, scheduling, and various other fields; see, for instance, the works by Steinberg [100], Elshafei [32], Eschermann and Wunderlich [33], and Bierwirth et al. [6]. For more extensive lists of applications of QAPs, we refer to the survey works by Pardalos et al. [80], Burkard et al. [18], Çela [21], Loiola et al. [66], and most recently Burkard et al. [17].

The quadratic assignment problem is typically described as a facility layout problem. In this model, it is assumed that there are a set of facilities F and a set of locations L of equal size. For each pair of facilities, a *flow* is specified via $a\colon F \times F \to \mathbb{R}$. Similarly, for each pair of locations, we specify a *distance* via $b\colon L \times L \to \mathbb{R}$. An assignment of the facilities to the locations is a bijection between the elements of the corresponding sets $\pi\colon F \to L$. The problem is to find an assignment π that minimizes the sum of the accompanied costs:

$$\min \sum_{i \in F} \sum_{j \in F} a(i,j) b(\pi(i), \pi(j)).$$

In this context, the flow and distance functions are usually viewed as real-valued matrices (a_{ij}) and (b_{ij}). The elements of these matrices are then indexed via the corresponding

element positions in F and L, respectively, leading to an equivalent formulation for the QAP:

$$\min \sum_{i=1}^{|F|} \sum_{j=1}^{|F|} a_{ij} b_{\pi(i)\pi(j)},$$

where $|F|$ denotes the cardinality of F, and π is a permutation over which we want to minimize the sum of the costs. Very often the objective is extended by another term that contains the initial costs for placing facility i at location $\pi(i)$. The introduction of this term gives

$$\min \left(\sum_{i=1}^{|F|} \sum_{j=1}^{|F|} a_{ij} b_{\pi(i)\pi(j)} + \sum_{i=1}^{|F|} c_{i\pi(i)} \right). \tag{1.1}$$

This is the model we will be working with in this thesis.

In order to show that the QAP is NP-hard, Sahni and Gonzalez [96] exploited the presentability of the traveling salesman problem (TSP) as a QAP. More precisely, they utilized their own argumentation for the NP-hardness of the ϵ-approximation problem for TSP to prove that the ϵ-approximation problem for QAP is NP-hard. Nowadays, the QAP is widely considered as one of the hardest NP-hard problems. The main reason for this can be found in the enormous difficulty of computing good quality bounds in branch-&-bound algorithms. High practical experience suggests that - despite the usage of modern computers - QAPs of size $n > 30$ are typically recognized as great challenging computational problems. The authors of [17, Chapter 8.1] write:

> All main algorithmic techniques for the exact solution of NP-hard problems have been used for attacking the QAP: decomposition, branch-&-bound, and branch-&-cut. The QAP, however, bravely resisted. After decades of attacks, the results obtained on its optimal solution are far from being satisfactory [...]

The situation is pretty much the same for related problems such as the *Bottleneck QAP*[1] [100] or the *Quadratic Semi-Assignment Problem* [42]. It is even worse for the *BiQuadratic Assignment Problem* [16], which is a generalization of the QAP. The bounding techniques for these problems are very similar to the ones developed for the QAP. Indeed, most of these techniques originate from corresponding procedures designed

[1] often also referred to as *Quadratic Bottleneck Assignment Problem*

for the quadratic assignment problem. This circumstance was a strong motivation for this thesis.

1.2. Notation and preliminary assumptions

In this work, we are concerned with Koopmans-Beckmann trace formulation of the quadratic assignment problem [31]:

$$\min_{X \in \Pi^n} \quad \operatorname{tr}(AXBX^T + CX^T), \tag{KBQAP}$$

where $A, B, C \in \mathbb{R}^{n \times n}$ are the parameter matrices of the QAP, Π^n denotes the set of $n \times n$ permutation matrices, and tr() terms the trace function. Unless otherwise stated, we assume that both matrices A and B are symmetric. Furthermore, without loss of generality, it is assumed that the diagonal elements of A and B are equal to zero. If this is not the case, then the corresponding costs can be shifted into the linear term by setting $C_{\text{new}} := C + \operatorname{diag}(A)\operatorname{diag}(B)^T$, where $\operatorname{diag}(A)$ denotes a column vector formed of the diagonal elements of A. The reformulated QAP satisfies the above condition and is equivalent to the original problem. Since any QAP of size $n \leqslant 3$ can be reduced to a linear assignment problem (LAP), we further assume $n \geqslant 4$. Throughout this paper, $A = \sum_{i=1}^{n} \mu_i p_i p_i^T$ and $B = \sum_{i=1}^{n} \lambda_i q_i q_i^T$ shall denote the eigenvalue decompositions of the symmetric matrices A and B, respectively.

For the designation of eigenvalues of arbitrary matrices, $\lambda(\cdot)$ shall denote the vector consisting of the respective eigenvalues. In the same manner, we use $\sigma(H)$ to denote the vector of singular values of some matrix H. By the respective arrow superscripts, we symbolize a presupposed ordering of these values, i.e. $\lambda^{\downarrow}$, $\sigma^{\downarrow}$ for non-ascending and $\lambda^{\uparrow}$, $\sigma^{\uparrow}$ for non-descending order.

In the following sections, we are frequently concerned with the minimization over some unitarily invariant matrix norm. Such norms are invariant under multiplications with unitary matrices from left and/or right. Arbitrary unitarily invariant matrix norms are identified by triple vertical lines, that is $|||\cdot|||$. Norms of particular interest are the spectral norm denoted by $|||\cdot|||_2$, the trace norm $|||\cdot|||_{\mathrm{T}}$ and the Frobenius norm $|||\cdot|||_{\mathrm{F}}$. The Euclidean norm to a vector $v \in \mathbb{R}^n$ is denoted by $\|v\|$.

Single vertical lines on both sides of a matrix are referring to the corresponding matrix of absolute values, hence $|H| = (|h_{ij}|)$. On the other hand, if Ω is a discrete set, then $|\Omega|$

shall denote its cardinality. The dimension of a vector space $\mathcal{V}$ describes the cardinality of a basis of $\mathcal{V}$. It is denoted by $\dim(\mathcal{V})$.

Additionally to the trace inner product of two real matrices $G, H \in \mathbb{R}^{m\times n}$ denoted by $\langle G, H\rangle := \operatorname{tr}(G^T H)$, we use $G \otimes H = (g_{ij}H) \in \mathbb{R}^{m^2\times n^2}$ and $G \circ H = (g_{ij}h_{ij}) \in \mathbb{R}^{m\times n}$ to term the Kronecker and Hadamard product of these matrices, respectively. Furthermore, we write $H^\dagger$ for the Moore-Penrose pseudoinverse of H [74, 85]. If H is an operator, $\mathcal{R}(H)$ denotes its range in the sense of its image. If H is a matrix, we use the same notation referring to its column space.

The cone of symmetric positive semidefinite matrices is of major importance for every discussion about semidefinite programming (SDP) relaxations. We denote the space of $n \times n$ symmetric matrices by $\mathcal{S}^n$ and its positive semidefinite subset by $\mathcal{S}^n_+$. In the same way, $\mathcal{S}^n_{++}$ denotes the cone of positive definite matrices in $\mathcal{S}^n$. In this context, we also use the relation signs $\{\succ, \succeq, \prec, \preceq\}$ to denote Löwner's partial ordering [48, 69]:

$$\forall H, G \in \mathcal{S}^n\colon \quad H \succeq G \iff H - G \in \mathcal{S}^n_+, \quad H \succ G \iff H - G \in \mathcal{S}^n_{++}.$$

The above notation is distinguished from a different utilization of the same symbols. If the compared objects are real vectors, $\{\succ, \prec\}$ are used to denote a preorder on these vectors. For two real vectors v, u of the same dimension, $u \prec v$ means that u is majorized by v. The majorization relation between these vectors is defined as

$$\forall u, v \in \mathbb{R}^n\colon \quad u \prec v \iff \sum_{i=1}^n u_i = \sum_{i=1}^n v_i \quad \text{and} \quad \sum_{i=1}^l u_i^\downarrow \leqslant \sum_{i=1}^l v_i^\downarrow \quad \text{for} \quad 1 \leqslant l < n,$$

where $u_i^\downarrow$ and $v_i^\downarrow$ denote the elements of u and v, respectively, rearranged in non-ascending order. Majorization is a preorder since it is a binary relation that is reflexive and transitive, but not antisymmetric.

Let $e_{(n)}$ denote the vector of all ones, $0_{(n)}$ the vector of all zeros, and $I_{(n)} = [e_1, \ldots, e_n]$ the $n \times n$ identity matrix. In addition to the already mentioned sets, we consider

- the space of $m \times n$ matrices: $\mathcal{M}^{m,n} = \mathbb{R}^{m\times n}$,
- the space of $n \times n$ matrices whose row and column sums are equal to one: $\mathcal{E}^n = \{M \in \mathcal{M}^{n,n} \mid Me_{(n)} = M^T e_{(n)} = e_{(n)}\}$,
- the cone of nonnegative matrices: $\mathcal{N}^{m,n} = \{M \in \mathcal{M}^{m,n} \mid \forall i, j\colon m_{ij} \geqslant 0\}$,
- the set of $n \times n$ double stochastic matrices: $\mathcal{D}^n = \mathcal{E}^n \cap \mathcal{N}^{n,n}$,

- the set of real matrices with orthonormal columns: $\mathcal{Q}^{m,n} = \{M \in \mathcal{M}^{m,n} \mid M^T M = I_{(n)}\}$.

Generally, we spare redundant informations on matrix dimensions. For instance, we write $\mathcal{M}^m$ and $\mathcal{N}^n$ instead of $\mathcal{M}^{m,m}$ and $\mathcal{N}^{n,n}$. Moreover, in cases where the dimension is evident from the context, the accompanying indicators may be discarded completely. For constants such as the all-ones vector e, the all-ones matrix $E = ee^T$, and the identity matrix I, whenever the dimension is not totally apparent from the context, the absence of these flags indicates dimension n and $n \times n$, respectively.

As already mentioned, $\mathrm{diag}(\cdot)$ denotes a linear operator that acts on a square matrix and gives the vector which consists of the diagonal elements of this matrix. The adjoint operator to $\mathrm{diag}(\cdot)$ is denoted by $\overset{*}{\mathrm{diag}}(\cdot)$. It transforms a vector into a diagonal matrix whose diagonal entries are identical to the respective vector elements. The successive application of both operators $\mathrm{diag}(\cdot)$ and $\overset{*}{\mathrm{diag}}(\cdot)$ resets all off-diagonal elements to zero and is abbreviated by $(\cdot)_{\mathrm{diag}} = \overset{*}{\mathrm{diag}}(\mathrm{diag}(\cdot))$. In this respect, H_{diag} denotes the diagonal matrix to H. Complementary to the diag-operator, $\mathrm{off}(H)$ denotes a column vector that contains all off-diagonal elements of the matrix H. This vector is obtained by vertical concatenation of the columns of H, but without its diagonal elements. Again, H_{off} abbreviates the successive application of $\mathrm{off}(\cdot)$ and its adjoint, such that $H_{\mathrm{off}} = H - H_{\mathrm{diag}}$. Other considered linear transformations are the full and the strictly lower triangular vectorization of a matrix. The former is denoted by $\mathrm{vec}(H)$ and describes the vector obtained by vertical concatenation of the columns of H. The latter is denoted by $\mathrm{tri}(H)$. The arranging rule is the same as for $\mathrm{off}(H)$ or $\mathrm{vec}(H)$, but the operator considers solely the strictly lower triangular elements of H (without matrix diagonal). We follow the typical notion for the vector to matrix transformation, and use $\mathrm{mat}(\cdot)$ to denote the adjoint operator to $\mathrm{vec}(\cdot)$. Naturally, the successive application of both operators gives the identity, such that $H = \mathrm{mat}(\mathrm{vec}(H))$ for any square matrix H.

In this thesis, the relation signs $\{<, \leqslant, >, \geqslant\}$ are used for the notation of element-wise inequalities. This convention induces a partial order in respect of the cone of nonnegative (positive) matrices, hence

$$\forall G, H \in \mathcal{M}^{m,n}\colon \quad G \geqslant H \iff G - H \in \mathcal{N}^{m,n}. \tag{1.2}$$

These inequality relations as well as the equality sign '=' may also be used in combination with the operators from above, for example $\{=_{\mathrm{diag}}, \geqslant_{\mathrm{diag}}, >_{\mathrm{diag}}, \ldots\}$. In case of the

subscript $_{\text{tri}}$, for instance, the respective relations apply only to the strictly lower triangular elements of the corresponding matrices, hence $G \geqslant_{\text{tri}} H$ is a short form for $\text{tri}(G) \geqslant \text{tri}(H)$.

Beyond the use as an adjoint to $\text{diag}(\cdot)$, we generalize the usage of $\overset{*}{\text{diag}}(\cdot)$ for the construction of block-diagonal matrices. The notation deviates from the previous one simply by the number of arguments. In this respect, $\overset{*}{\text{diag}}(H_1, H_2, \ldots, H_m)$ denotes the block-diagonal matrix which consists of the corresponding block matrices $H_1, \ldots, H_m$. For these matrices, it is not required that they are square or that they have the same dimension.

1.3. Outline and contributions

One of the main motivations for this thesis lies in the enormous difficulty of computing good quality bounds for the QAP. Driven by this motivation, we investigated various bounding techniques. In particular, we researched different and derived new relaxation strategies which are individually designed for the quadratic assignment problem. We found some concepts to improve these relaxations and developed new relaxation strategies on the basis of well-known bounding procedures, recent developments in this field, and our own ideas.

In this work, our concern is the computation of lower bounds via SDP relaxations for QAP instances which are representable in the form (KBQAP) and satisfy the assumption that A and B are symmetric. In order to come up with tight, numerically stable and efficient relaxation frameworks, we have investigated many different relaxation approaches, tested various formulations, and performed tests with a large number of additional constraints. Many of our approaches were misdirected or came to nothing, and it would be far beyond the scope of this thesis to explain all of them. Nevertheless, in the attempt to spare the inclined reader and researcher going through the same considerations again, we often include the process of decisions-shaping into our explanations. Sometimes we even include intermediate development steps from the underlying conceptional idea to the final realization. By doing so, we are aiming for a better comprehensibility. The strong focus on the derivation of the corresponding bounding concepts and the emphasis on connections between different approaches lead to smooth transitions from results known in literature to our own developments. Unfortunately, this sometimes tends to obscure the distinctions of our contributions. Not every result that we introduce in this thesis is explicitly marked as our own. However, all known results in the literature are

clearly exposed as such. If no dependence is mentioned when the respective result is introduced, then it refers to our own research. For a better distinction of the contributions in this thesis, subsequently, we give a small sum up of its contents.

Chapter 2 recaps some well-known, fundamental theorems which are essential for the understanding of the subsequent explanations. This comprises, for instance, the Schur complement condition for semidefiniteness, or Birkhoff's theorem for doubly stochastic matrices. The former result is elementary for every discussion about semidefinite relaxations for non-convex quadratic programming problems. The latter is crucial for linearizations of assignment problems. This chapter also contains some auxiliary lemmas. First and foremost, these lemmas serve to simplify the proofs of subsequent results. Some of these auxiliary results, such as Lemma 2.15 and Lemma 2.16, also help to understand fundamental connections between related programming problems which are discussed in subsequent chapters.

In Chapter 3, we sum up and compare four basic relaxation concepts for the quadratic assignment problem. These concepts comprise the Vector Lifting (*VL*) approach [86, 108, 116], the Matrix Lifting (*ML*) technique [27], the Matrix Splitting (*MS*) procedure [83,84] and the "eigenspace" SDP (*ES*) relaxation [25], here listed in chronological order. The corresponding relaxations are explained in Section 3.1. In Section 3.2, we show that there is an ordering of the corresponding lower bounds which is in concurrence with the complexity of the respective relaxation. With $\mathcal{O}(n^4)$ variables and $\mathcal{O}(n^3)$ equality constraints, the relaxation based on the vector lifting approach is the most expensive one. In return, however, this relaxation is provably tighter than its competitors. The second in line, with $\mathcal{O}(n^3)$ variables and $\mathcal{O}(n^2)$ equalities, is the "eigenspace" SDP relaxation. We show the superiority of this relaxation in comparison to the frameworks based on the matrix lifting technique and the matrix splitting procedure. The latter relaxations have both $\mathcal{O}(n^2)$ variables and $\mathcal{O}(n^2)$ equality constraints. The relaxation based on matrix lifting is slightly more expensive than the one based on matrix splitting, but there is no ordering between them.

In the first part of Chapter 4, we explain and discuss different reformulation strategies for the actual assignment problem. A QAP instance specified by the data vector $(\acute{A}, \acute{B}, \acute{C})$ is a reformulation of a given QAP instance (A, B, C) if there exists a bijection $\mathbf{X}\colon \Pi^n \to \Pi^n$ satisfying

$$\forall X \in \Pi^n\colon \qquad \operatorname{tr}(\acute{A}X\acute{B}X^T + \acute{C}X^T) = \operatorname{tr}(A\mathbf{X}(X)B\mathbf{X}(X)^T + C\mathbf{X}(X)^T).$$

A reformulation strategy of particular interest is investigated in Subsection 4.1.1. The corresponding transformations are referred to as QAP reformulations and can be described via six n-component vectors d_A, d_B, v_A, v_B, w_A, and w_B. Our contribution to this topic is the transfer and adaptation of this reformulation technique - which is well-known and often used to obtain tighter QAP linearizations - for the use in combination with the respective SDP relaxations. For this purpose, we first show that only two of the parameter vectors, d_B and v_B, can be chosen independently and are actually affecting the discussed SDP relaxations. The class of equivalent QAP formulations which are of interest for the considered relaxations can thereby be described via

$$\forall d_B, v_B \in \mathbb{R}^n: \quad (A, B, C) \sim (\acute{A}, \acute{B}, \acute{C}) = (A, B + \overset{*}{\text{diag}}(d_B) + v_B e^T + e v_B^T, C - 2\acute{A} e v_B^T),$$

where $\sim$ is used to denote the equivalence of the instances (A, B, C) and $(\acute{A}, \acute{B}, \acute{C})$ in regard to their objective terms and the corresponding optimal permutations. Subsequently, we explain different strategies to attain appropriate representatives of d_B and v_B. We discuss different optimization criteria and give strong arguments for our final choice of parameter vectors. The actually used QAP reformulation is defined in (4.20).

The second part of Chapter 4 is not about reformulations of the actual QAP but about reformulation strategies for the respective relaxation frameworks discussed in the previous chapter. In Section 4.2, we describe possible reductions of the number of equality constraints by substituting a smaller number of equivalent conditions for them. The chapter is concluded with a small summary of the applied reformulation techniques and the presentation of the level-1 versions of the discussed relaxations. In this context, it should be mentioned that the SDP frameworks are not only classified with respect of the underlying relaxation concept (see above) but also in consideration of the level of modifications. Level-0 refers to SDP programs which are plain realizations of the corresponding relaxation concept. Neither reformulations nor cutting techniques from any other relaxation concept are applied. Level-1 relaxations are the reformulated versions of the SDP frameworks. The applied modifications are explained in Section 4.3. Level-2 refers to the relaxation instances which apply both the reformulation strategies discussed in Chapter 4 and the cuts presented in Chapter 6. Another property of the level-1 frameworks is the use of the same matrix variables $X \in \mathcal{D}^n$ and $Y \in \mathcal{S}^n$ independently of the underlying relaxation concept. Here the symmetric matrix variable Y is always used to relax the quadratic term XBX^T. The general utilization of these variables unifies and thereby simplifies the incorporation of additional constraints. Furthermore, this leads to

the same objective function in all presented level-1 relaxations:

$$\operatorname{tr}(AY + CX^T) = \langle A, Y\rangle + \langle C, X\rangle.$$

Considering the relation between Y and XBX^T, the above objective function is evidently the counterpart to the objective term of (KBQAP).

The main contributions in this thesis are explained in Chapter 5. There we introduce three new relaxation programs. The first program, which we refer to as *ESC*, is derived from the "eigenspace" SDP relaxation by approximating the corresponding eigenspace via some clustering algorithm. In return for a moderate widening of the feasible set, the eigenspace clustering usually leads to a significantly more economical framework. The second relaxation presented in this chapter is referred to as *MLX*. This relaxation is based on the concept of matrix lifting and describes an extension of *ML*. In Subsection 5.1.2, we explain how *MLX* is constructed and validate its superiority compared to *ML*. The third newly introduced framework is based on a concept that we call *inverse interrelated matrix splitting*. As the name suggests, this concept is closely related to the matrix splitting approach. To be more specific, it describes an extension of the idea of positive semidefinite matrix splitting by an inverse relation between the respective splitting parts. The corresponding framework is explained in Subsection 5.1.3. It is referred to as *IIMS*. The chapter is concluded with a short comparison between bounds obtained by solving the new relaxation programs *ESC*, *MLX*, *IIMS*, and the results computed via their origins *ES*, *ML*, and *MS*, respectively. In Corollary 5.4, we show orderings between these bounds.

In Chapter 6, we explain the incorporation of additional linear programming (LP) and second order cone programming (SOCP) constraints by which we derive the level-2 versions of the respective relaxation frameworks. These constraints are called cuts because they originate in other relaxation concepts and are used to cut away parts of the feasible set connected with the matrix variable Y. In Section 6.1, we are concerned with four classes of cuts: Gilmore-Lawler bound (*GLB*) based inequalities, constraints which originate from eigenvalue bound (*EVB*) majorizations, linear inequalities based on symmetric functions, and convex p-norm inequalities. Incorporations of the latter three types of constraints are known in literature; however, the Gilmore-Lawler bounding procedure has not been combined with an SDP relaxation before. Besides the incorporation of *GLB* based constraints, our contribution to this topic includes various improvements of the different cutting strategies. We introduce improved upper bounds for an eigenvalue related

bounding technique by Xia [110] and demonstrate how this concept can be integrated into the respective SDP relaxations. Subsection 6.1.3 is used to link the minimum and maximum bounds introduced by Mittelmann and Peng [73] with the corresponding sum-matrix inequalities. This connection is then exploited for the construction of tighter versions of these bounds. In Subsection 6.1.4, we then introduce a provably stronger version of the norm constraints used in [83], [84], and [25]. The discussion about cutting strategies is completed by giving an appropriate selection of constraints for the different types of presented SDP relaxations.

In order to fill the gap between the theoretical relaxation concepts and the practical computation of lower bounds, we discuss various questions regarding the actual realization of the presented frameworks. This is done in Chapter 7. Since the investigation of problem representations and implementation details typically offers few interesting and/or new insights on the actual topic, we begin this chapter by motivating the research of implementation issues. For this purpose, we give numerical results to demonstrate the advantage of individually adapted realizations over straightforward implementations. The second part of this chapter is about the disclosure and the resolution of numerical difficulties accompanied by the respective implementations. By Theorem 7.1, we reveal the alarming situation that all the previously discussed SDP relaxations for the QAP are ill-posed. Subsequently, we recap two different approaches to attack this problem. One of these approaches, the facial reduction, is known since the early 80's when Borwein and Wolkowicz introduced the corresponding procedures [10,11]. It was already applied to the vector lifting based SDP relaxation in 1998 by Zhao, Karisch, Rendl, and Wolkowicz [116]. Our contribution to this topic is the adaptation for the other SDP frameworks and, more importantly, the modifications described in Subsection 7.2.4 which lead to the practical applicability of the facial reduction procedure. After presenting a way to compute verified bounds for the optimal objective value to the respective relaxation instances, we discuss a few more beneficial remarks on implementation details. We conclude this chapter by evaluating numerical results for level-2 relaxations of a wide range of instances from the QAP library [18].

Chapter 8 concludes this thesis with a short summary of the presented developments. There we also evaluate strengths and weaknesses of the presented level-2 relaxation frameworks. Additionally, the chapter serves the discussion of future directions.

In Appendix A, we give a quick reference for the Matlab/Octave software package which originated in the context of this thesis.

Chapter 2.

Preparations

2.1. Auxiliary results and further preparations

This section is used to present some auxiliary lemmas. We begin by stating some widely known, fundamental theorems. In consideration of the scope of this work, we believe that a quick recap of these fundamental results pays off when working through the topics of the following sections. The respective theorems can be found, for instance, in Bhatia's [5] or Horn and Johnson's [48].

The most essential utility for the semidefinite relaxation of quadratic constraints is the Schur complement condition for (semi)definiteness. In all discussed relaxation frameworks, we make use of its generalization in terms of pseudoinverses.

Theorem 2.1 (Albert [1])**.** *Let $H_{11} \in \mathcal{S}^m$, $H_{22} \in \mathcal{S}^n$ and $H_{12} \in \mathcal{M}^{m,n}$ be matrix blocks of a symmetric $(m+n) \times (m+n)$ matrix H, arranged as*

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{bmatrix}.$$

Then $H \succeq 0$ if and only if $H_{11} \succeq 0$, $H_{22} - H_{12}^T H_{11}^\dagger H_{12} \succeq 0$, and $H_{11} H_{11}^\dagger H_{12} = H_{12}$. Similarly, the condition $H \succ 0$ is equivalent to $H_{11} \succ 0$, $H_{22} - H_{12}^T H_{11}^{-1} H_{12} \succ 0$.

For a detailed review of the history and many applications of the Schur complement, we refer to [114].

Another famous and fundamental result was given by von Neumann [105]. Ahead of stating his result, let us recall what kind of functions are denoted as symmetric gauge functions.

Definition 2.2. *A norm* $\mathrm{g}\colon \mathbb{R}^n \to \mathbb{R}_+$ *is said to be permutation invariant or symmetric if*

$$\forall v \in \mathbb{R}^n, X \in \Pi^n\colon \quad \mathrm{g}(v) = \mathrm{g}(Xv). \tag{2.1}$$

The norm is called gauge invariant or absolute if it satisfies

$$\forall v \in \mathbb{R}^n\colon \quad \mathrm{g}(v) = \mathrm{g}(|v|). \tag{2.2}$$

A permutation and gauge invariant norm is called a symmetric gauge function.

Theorem 2.3 (von Neumann [105])**.** *Any unitarily invariant matrix norm is a symmetric gauge function of the singular values of the respective matrix.*

In consideration of this connection, we say that a unitarily invariant matrix norm is strictly monotone whenever this statement applies to the corresponding symmetric gauge function, i.e.

$$\forall G, H \in \mathcal{M}^{m,n}\colon \quad \sigma(G) \leqslant \sigma(H),\, \sigma(G) \neq \sigma(H) \implies |||G||| < |||H|||. \tag{2.3}$$

The following well-known result is referred to as Eckart-Young-Mirsky theorem.

Theorem 2.4 (Mirsky [72])**.** *Denote by* $H = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T$ *the singular value decomposition of some matrix* $H \in \mathcal{M}^{m,n}$*, and assume that the singular values* σ_i *are ordered in non-ascending order. For some natural number* r *not greater than the rank of* H*, consider the approximation problem*

$$\inf_{G \in \mathcal{M}^{m,n}} \quad \{|||H - G|||\colon \operatorname{rank}(G) \leqslant r\}. \tag{2.4}$$

Regardless of the choice of the unitarily invariant matrix norm $|||\cdot|||$*,*

$$\hat{G} = \sum_{i=1}^{r} \sigma_i u_i v_i^T \tag{2.5}$$

is a solution to problem (2.4)*.*

The low-rank matrix approximation approach from above was introduced by Eckart and Young [30]. They proved the optimality of (2.5) for the Frobenius norm. Mirsky [72, Theorem 2] generalized their result for arbitrary unitarily invariant matrix norms.

Theorem 2.5 (Lidskiĭ [63])**.** *For any two symmetric matrices G and H of the same dimension, the relation*

$$\lambda^{\downarrow}(G+H) - \lambda^{\downarrow}(G) \prec \lambda(H). \tag{2.6}$$

holds valid.

Theorem 2.6 (Schur [97])**.** *The vector consisting of the diagonal elements of a symmetric matrix is majorized by the vector that contains its eigenvalues, i.e.*

$$\forall H \in \mathcal{S}\colon \quad \operatorname{diag}(H) \prec \lambda(H). \tag{2.7}$$

The inspection of different related relaxation programs reveals that permutation matrices are typically relaxed by doubly stochastic matrices. This also applies to the semidefinite programming relaxations that will be discussed in the following sections. For this reason, we gather some essential properties of doubly stochastic matrices.

Theorem 2.7 (Birkhoff [7])**.** *The set of doubly stochastic matrices is identical with the convex hull of the set of permutation matrices.*

Theorem 2.8 (Hardy, Littlewood and Pólya [45])**.** *A vector u is majorized by another vector v if and only if there is a doubly stochastic matrix S that transforms v into u, i.e.*

$$u \prec v \iff \exists S \in \mathcal{D}\colon u = Sv. \tag{2.8}$$

Corollary 2.9. *Given a pair of symmetric matrices G and H of the same dimension, there exists a doubly stochastic matrix S such that*

$$\lambda^{\downarrow}(G+H) - \lambda^{\downarrow}(G) = S\lambda(H). \tag{2.9}$$

Proof. The result follows immediately by the eigenvalue inequality given in Theorem 2.5 and the equivalence stated in Theorem 2.8. □

In the second part of this section, we give some auxiliary lemmas which will be utilized in the proofs of the subsequent results.

Lemma 2.10. *For given vectors $v, w \in \mathbb{R}^n$ with nonnegative Hadamard product $v \circ w \geqslant 0$ and a symmetric gauge function $\mathrm{g}\colon \mathbb{R}^{2n} \to \mathbb{R}_+$, it is*

$$\mathrm{g}\left(\begin{pmatrix} v - w \\ 0 \end{pmatrix}\right) \leqslant \mathrm{g}\left(\begin{pmatrix} v \\ w \end{pmatrix}\right). \tag{2.10}$$

Proof. For any $i \in \{1, \ldots, n\}$, the nonnegativity condition on the Hadamard product gives $v_i w_i \geqslant 0$ and thereby $|v_i - w_i| \leqslant \max\{|v_i|, |w_i|\}$. Hence, there exists a permutation matrix $X \in \Pi^{2n}$ satisfying

$$\left|\begin{pmatrix} v - w \\ 0 \end{pmatrix}\right| \leqslant X \left|\begin{pmatrix} v \\ w \end{pmatrix}\right|.$$

The remainder of the argument follows by Definition 2.2. □

Lemma 2.11. *Given a symmetric matrix H, the relation*

$$|||H||| \geqslant |||(H)_{\mathrm{diag}}||| \tag{2.11}$$

holds valid for arbitrary unitarily invariant matrix norms $|||\cdot|||$.

Proof. By Theorem 2.6 and 2.8, we conclude the existence of some doubly stochastic matrix S such that

$$\mathrm{diag}(H) = S\lambda(H).$$

Due to Birkhoff's theorem, it is moreover apparent that S is a convex combination of permutation matrices, therefore

$$\exists \alpha_i \in \mathbb{R}_+, X_i \in \Pi\colon \quad S = \sum_{i=1}^{|\Pi|} \alpha_i X_i, \ \sum_{i=1}^{|\Pi|} \alpha_i = 1,$$

where Π denotes the set of all permutation matrices of proper dimension, and $|\Pi|$ is used to state its cardinality. For the diagonal vector of H, this gives the identity

$$\mathrm{diag}(H) = \sum_{i=1}^{|\Pi|} \alpha_i X_i \lambda(H).$$

Since $|||\cdot|||$ is unitarily invariant, we further have $|||H||| = ||| \operatorname{diag}^*(\lambda(H))||| = ||| \operatorname{diag}^*(X_i\lambda(H))|||$ for all $X_i \in \Pi$ of proper dimension. Taken together, this gives

$$|||(H)_{\mathrm{diag}}||| = ||| \sum_{i=1}^{|\Pi|} \alpha_i \operatorname{diag}^*(X_i\lambda(H))||| \leqslant \sum_{i=1}^{|\Pi|} \alpha_i ||| \operatorname{diag}^*(X_i\lambda(H))||| = |||H|||.$$

□

Lemma 2.12. *Let G, H be symmetric positive semidefinite matrices of the same dimension. The implication*

$$G \succeq H \succeq 0 \quad \Longrightarrow \quad |||G||| \geqslant |||H||| \tag{2.12}$$

is valid for any unitarily invariant matrix norm $|||\cdot|||$. If, in addition, the considered norm is strictly monotone, then it satisfies the stronger implication

$$G \succeq H \succeq 0,\, G \neq H \quad \Longrightarrow \quad |||G||| > |||H|||. \tag{2.13}$$

Proof. The result follows straightforwardly from von Neumann's theorem and Lidskiĭ's inequality. Since $G - H$ is symmetric positive semidefinite, it is $S\lambda(G-H) \geqslant 0$ for any doubly stochastic matrix S of proper size. By Corollary 2.9, we have $\lambda^{\downarrow}(G) \geqslant \lambda^{\downarrow}(H) \geqslant 0$. Together with the symmetric positive semidefiniteness of the matrices G and H this gives $\sigma^{\downarrow}(G) \geqslant \sigma^{\downarrow}(H)$. Equation (2.12) is then an immediate consequence of Theorem 2.3. For $G \neq H$, we necessarily have $\operatorname{tr}(G) > \operatorname{tr}(H)$ and thereby $\sigma^{\downarrow}(G) \neq \sigma^{\downarrow}(H)$. By definition, any strictly monotone symmetric unitarily invariant matrix norm therefore satisfies the strict inequality in (2.13). □

Lemma 2.13. *Let G and H be symmetric positive semidefinite matrices of the same dimension. The validity of Löwner's partial ordering relation $G \succeq H$ implies that the column space of H is included in the column space of G.*

Proof. For the following reductio ad absurdum, assume the existence of a nontrivial vector v that lies in $\mathcal{R}(G)^{\perp}$ but not in the orthogonal complement of $\mathcal{R}(H)$. Naturally, such a vector has a nonzero part in the column space of H, i.e.

$$v = v_1 + v_2, \qquad 0 \neq v_1 \in \mathcal{R}(H),\, v_2 \in \mathcal{R}(H)^{\perp}.$$

By $v_1 \in \mathcal{R}(H)$, it is evident that v_1 can be represented as a linear combination of eigenvectors of H which correspond to strictly positive eigenvalues. Hence, $v_1 \neq 0$ implies

$v_1^T H v_1 > 0$ and gives

$$v^T H v = v_1^T H v_1 > 0 = v^T G v,$$

resulting in a contradiction to the requirement $G \succeq H$. It follows $\mathcal{R}(G)^\perp \subseteq \mathcal{R}(H)^\perp$ which is equivalent to the statement $\mathcal{R}(G) \supseteq \mathcal{R}(H)$. □

Lemma 2.14. *Let H be a symmetric positive semidefinite matrix. For any normalized vector u that lies in the column space of H, i.e. $\|u\| = 1$ and $u \in \mathcal{R}(H)$, it is*

$$u^T H^\dagger u \geqslant (u^T H u)^{-1}. \tag{2.14}$$

Proof. The Moore-Penrose pseudoinverse of H satisfies the identity $H^\dagger H H^\dagger = H^\dagger$. This and the existence of $H^{\frac{1}{2}}$ shows

$$\begin{bmatrix} u^T H^\dagger u & u^T H^\dagger H u \\ u^T H H^\dagger u & u^T H u \end{bmatrix} = \begin{bmatrix} u^T H^\dagger H^{\frac{1}{2}} \\ u^T H^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} u^T H^\dagger H^{\frac{1}{2}} \\ u^T H^{\frac{1}{2}} \end{bmatrix}^T \succeq 0.$$

The positive semidefinitenes of this matrix implies a non-negative determinant and therefore $(u^T H^\dagger u)(u^T H u) - (u^T H^\dagger H u)^2 \geqslant 0$. The matrix $H^\dagger H$ states an orthogonal projection matrix for the column space of H. Any vector that lies in this space is unaffected by the multiplication with this projection matrix, such that $u^T H^\dagger H u = u^T u = 1$. By the above determinant inequality, we then derive $(u^T H^\dagger u)(u^T H u) \geqslant 1$, which completes the proof. □

Lemma 2.15. *For all symmetric positive semidefinite matrices G and any doubly stochastic matrix S of proper dimension, there exists another symmetric positive semidefinite matrix H that satisfies the three conditions*

$$H \succeq S G S^T, \quad \operatorname{diag}(H) = S \operatorname{diag}(G), \quad \textit{and} \quad He = SGe. \tag{2.15}$$

Proof. Most of the arguments for Lemma 2.15 are borrowed from the proof of [84, Theroem 1] by Peng, Zhu, Luo, and Toh. We follow their line of argument. Since S is doubly stochastic, it may be expressed as a convex combination of permutation matrices, i.e. $S = \sum_{i=1}^{|\Pi|} \alpha_i X_i$ for some $\alpha_i \in \mathbb{R}_+$ and $X_i \in \Pi$ such that $\alpha_1 + \alpha_2 + \ldots + \alpha_{|\Pi|} = 1$. The

authors of [84] defined the specific matrix

$$H := \sum_{i=1}^{|\Pi|} \alpha_i X_i G X_i^T,$$

and showed that it satisfies all conditions in (2.15). For every permutation matrix X_i, it is $\operatorname{diag}(X_i G X_i^T) = X_i \operatorname{diag}(G)$ and $X_i G X_i^T e = X_i G e$. This holds necessarily valid for any linear combination of these identities. Thus, H satisfies the equality conditions in (2.15). For the validation of the semidefiniteness condition, one can exploit the facts that H is a convex combination of the matrices $\{X_i G X_i^T\}$, and that - by positive semidefiniteness of G - the function $\mathrm{f}_G(v) = v^T G v$ is convex. For any real vector v of proper dimension, it is

$$v^T H v = \sum_{i=1}^{|\Pi|} \alpha_i \, \mathrm{f}_G(X_i v) \geqslant \mathrm{f}_G(\sum_{i=1}^{|\Pi|} \alpha_i X_i v) = \mathrm{f}_G(Sv) = v^T S G S^T v$$

and therefore $H \succeq S G S^T$. □

The result of Lemma 2.15 is not only interesting in the context of some proofs, it also pinpoints the fact that the described SDP relaxations have feasible points for every choice of $X \in \mathcal{D}$.

Lemma 2.16. *For given $B \in \mathcal{S}^n$ with eigenvalue decomposition $B = \sum_{i=1}^n \lambda_i q_i q_i^T$, a fixed parameter vector $\zeta \in \mathbb{R}^n$, and nonnegative coefficients $w_1, w_2 \in \mathbb{R}_+$, consider the minimization problem*

$$\begin{aligned} \inf_{G,H \in \mathcal{S}^n} \quad & w_1 |\!|\!| H |\!|\!| + w_2 |\!|\!| H - B |\!|\!| \\ \text{s. t.} \quad & \begin{bmatrix} H & G \\ G & H - B \end{bmatrix} \succeq 0, \\ & q_i^T G q_i = \zeta_i \qquad \text{for} \quad i \in \{1 \ldots, n\}. \end{aligned} \tag{2.16}$$

The norms in the objective function of problem (2.16) *are supposed to be unitarily invariant and may be chosen individually for each term. The following statements hold true:*

(*a*) *The matrix pair $(\hat{G}, \hat{H})$,*

$$\hat{G} := \sum_{i=1}^{n} \zeta_i q_i q_i^T, \qquad \hat{H} := \frac{1}{2} B + \sqrt{\frac{1}{4} B^2 + \hat{G}^2}, \tag{2.17}$$

defines a solution to this problem.

(*b*) *If the matrix norms are strictly monotone, the coefficients satisfy* $w_1 + w_2 > 0$*, and problem* (2.16) *is extended by the additional equality conditions*

$$\mathcal{G} q_i = \zeta_i q_i \qquad \textit{for} \quad i \in \{k \mid \lambda_k = 0, \zeta_k \neq 0\}, \tag{2.18}$$

then the optimal point in (2.17) *is unique.*

Proof. Define the symmetric orthogonal matrix

$$U := I - \hat{G}^\dagger \left(\hat{G} + \sqrt{\hat{G}^2} \right) = \sum_{i=1}^{n} \left(1 - \zeta_i^\dagger \max\{2\zeta_i, 0\} \right) q_i q_i^T.$$

If $\hat{G}$ has full rank, $\hat{G}^\dagger \hat{G}$ is the identity matrix. Otherwise, $\hat{G}^\dagger \hat{G}$ states an orthogonal projection matrix for the column space of $\hat{G}$. For both cases, we derive

$$U\sqrt{\hat{G}^2} = \sqrt{\hat{G}^2} - \hat{G}^\dagger \hat{G} \sqrt{\hat{G}^2} + \hat{G}^\dagger \hat{G}^2 = \hat{G}^\dagger \hat{G}^2 = \hat{G}.$$

Since the matrices $\hat{H}$, B, and U are all three simultaneously diagonalizable with the orthonormal eigenvector basis $\{q_i\}$, they commute, such that $\hat{H}^{\frac{1}{2}} U (\hat{H} - B)^{\frac{1}{2}} = U \hat{H}^{\frac{1}{2}} (\hat{H} - B)^{\frac{1}{2}}$. This product, in turn, satisfies

$$\begin{aligned} U\hat{H}^{\frac{1}{2}} (\hat{H} - B)^{\frac{1}{2}} &= U \frac{1}{\sqrt{2}} \left(B + \sqrt{B^2 + 4\hat{G}^2} \right)^{\frac{1}{2}} \frac{1}{\sqrt{2}} \left(-B + \sqrt{B^2 + 4\hat{G}^2} \right)^{\frac{1}{2}} \\ &= U \frac{1}{2} \left(-B^2 + B^2 + 4\hat{G}^2 \right)^{\frac{1}{2}} = U\sqrt{\hat{G}^2} = \hat{G}, \end{aligned}$$

which is used to show the nonnegative definiteness of

$$\begin{bmatrix} \hat{H} & \hat{G} \\ \hat{G} & \hat{H} - B \end{bmatrix} = \begin{bmatrix} \hat{H}^{\frac{1}{2}} U \\ (\hat{H} - B)^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \hat{H}^{\frac{1}{2}} U \\ (\hat{H} - B)^{\frac{1}{2}} \end{bmatrix}^T \in \mathcal{S}_+^{2n}.$$

This and the identities $q_i^T \hat{G} q_i = \zeta_i$ imply that $(\hat{G}, \hat{H})$ states a feasible pair of matrices to problem (2.16).

Let (G, H) denote an arbitrary feasible matrix pair to the considered minimization problem. For any eigenvector q_i of B, the positive semidefiniteness of

$$\begin{bmatrix} q_i & 0 \\ 0 & q_i \end{bmatrix}^T \begin{bmatrix} \mathsf{H} & \mathsf{G} \\ \mathsf{G} & \mathsf{H} - B \end{bmatrix} \begin{bmatrix} q_i & 0 \\ 0 & q_i \end{bmatrix} = \begin{bmatrix} q_i^T \mathsf{H} q_i & \zeta_i \\ \zeta_i & q_i^T \mathsf{H} q_i - \lambda_i \end{bmatrix}$$

implies a nonnegative determinant, thereby $(q_i^T \mathsf{H} q_i)(q_i^T \mathsf{H} q_i - \lambda_i) - \zeta_i^2 \geqslant 0$ for all $i \in \{1, \ldots, n\}$. By the solutions of the corresponding quadratic equalities and the nonnegativeness of $q_i^T \mathsf{H} q_i$, one derives the equivalent conditions

$$q_i^T \mathsf{H} q_i \geqslant \frac{\lambda_i}{2} + \sqrt{\left(\frac{\lambda_i}{2}\right)^2 + \zeta_i^2} = q_i^T \hat{H} q_i \qquad \text{for} \quad i = 1, 2, \ldots, n. \tag{2.19}$$

Together with the orthogonality of $Q := [q_1, q_1, \ldots, q_n]$, Lemma 2.11 gives

$$|||\mathsf{H}||| = |||Q^T \mathsf{H} Q||| \geqslant |||(Q^T \mathsf{H} Q)_{\mathrm{diag}}||| \geqslant |||(Q^T \hat{H} Q)_{\mathrm{diag}}||| = |||\hat{H}|||$$

and, by the same argument, $|||\mathsf{H} - B||| \geqslant |||\hat{H} - B|||$.

Hence, $(\hat{G}, \hat{H})$ defines a feasible pair of matrices which accompanies a minimal objective value. This completes the proof of statement (a).

If the norms are strictly monotone and $\mathrm{diag}(Q^T \mathsf{H} Q) \neq \mathrm{diag}(Q^T \hat{H} Q)$, then (2.19) implies $|||(Q^T \mathsf{H} Q)_{\mathrm{diag}}||| > |||(Q^T \hat{H} Q)_{\mathrm{diag}}|||$ as well as $|||(Q^T (\mathsf{H} - B) Q)_{\mathrm{diag}}||| > |||(Q^T (\hat{H} - B) Q)_{\mathrm{diag}}|||$. Since at least one of the coefficients w_1 or w_2 is nonzero, this requires that the vector equality $\mathrm{diag}(Q^T \mathsf{H} Q) = \mathrm{diag}(Q^T \hat{H} Q)$ holds valid whenever H corresponds to an optimal solution point.

In the remainder of the proof, it will be shown that every feasible matrix pair (G, H) satisfying (2.18) as well as $\mathrm{diag}(Q^T \mathsf{H} Q) = \mathrm{diag}(Q^T \hat{H} Q)$ is necessarily simultaneously diagonalizable by Q, i.e. the eigenvalue equations $\mathsf{G} q_i = \zeta_i q_i$ and $\mathsf{H} q_i = \lambda_i(\mathsf{H}) q_i$ are satisfied for all $i \in \{1, \ldots, n\}$. For each index i, we distinguish three cases subdividing the corresponding eigenvalue equations in regard to the zero property of the respective eigenvalue λ_i and the parameter ζ_i.

`Case 1.` Suppose that $\lambda_i = 0$ and $\zeta_i \neq 0$. Then, the corresponding constraint in (2.18) is active, and $q_i^T H q_i = q_i^T (H - B) q_i|_{\lambda_i=0} = q_i^T \hat{H} q_i|_{\lambda_i=0} = |\zeta_i|$ implies

$$\begin{pmatrix} -\zeta_i q_i \\ |\zeta_i| q_i \end{pmatrix}^T \begin{bmatrix} H & G \\ G & H - B \end{bmatrix} \begin{pmatrix} -\zeta_i q_i \\ |\zeta_i| q_i \end{pmatrix} = 2\zeta_i \left(\zeta_i q_i^T H q_i - |\zeta_i| q_i^T G q_i \right) = 0.$$

This gives us a vector in the kernel of this matrix:

$$\begin{bmatrix} H & G \\ G & H - B \end{bmatrix} \begin{pmatrix} -\zeta_i q_i \\ |\zeta_i| q_i \end{pmatrix} = 0 \quad \Longleftrightarrow \quad H q_i = \frac{\zeta_i}{|\zeta_i|} G q_i.$$

By (2.18), the left-hand side is identical to $|\zeta_i| q_i$.

`Case 2.` If ζ_i equals zero, the identity

$$q_i^T H q_i = q_i^T \hat{H} q_i|_{\zeta_i=0} = \frac{\lambda_i}{2} + \sqrt{\left(\frac{\lambda_i}{2}\right)^2} = \max\{\lambda_i, 0\}$$

implies that either $q_i^T H q_i = 0$ or $q_i^T (H - B) q_i = 0$. Together with the positive semidefiniteness of the matrices H and $H - B$, we derive either

$$H q_i = 0 \qquad \text{or} \qquad H q_i = B q_i = \lambda_i q_i.$$

Moreover, from the respective nullspace identity of the positive semidefinite block matrix, it follows $G q_i = 0$.

`Case 3.` Finally, suppose that $\lambda_i \neq 0$ and $\zeta_i \neq 0$. For the validation of the corresponding eigenvalue equations, we define the factor

$$\xi_i := \frac{2\zeta_i}{\lambda_i + \sqrt{\lambda_i^2 + 4\zeta_i^2}}.$$

The identity $\xi_i^2 q_i^T H q_i = \xi_i \zeta_i = q_i^T H q_i - \lambda_i$ gives

$$\begin{pmatrix} \xi_i q_i \\ -q_i \end{pmatrix}^T \begin{bmatrix} H & G \\ G & H - B \end{bmatrix} \begin{pmatrix} \xi_i q_i \\ -q_i \end{pmatrix} = 0.$$

Due to positive semidefiniteness of this block matrix, the vector necessarily lies in the corresponding kernel, resulting in the identities

$$\xi_i H q_i = G q_i \qquad \text{and} \qquad \xi_i G q_i = (H - B) q_i = H q_i - \lambda_i q_i.$$

Via substitution, we derive the desired eigenvalue equations $G q_i = \frac{\lambda_i \xi_i}{1-\xi_i^2} q_i = \zeta_i q_i$ and $H q_i = \frac{\lambda_i}{1-\xi_i^2} q_i$.

Taken together, this proves a unique eigenvalue decomposition of both solution matrices G and H, which finishes the argument for uniqueness of the optimal matrix pair. □

Chapter 3.

Semidefinite Programming Relaxations

3.1. Relaxation concepts

The concept of relaxations is a fundamental approach for the computation of lower or upper bounds of intractable programming problems. It can be used directly as an approximation of the original problem, for bound computations in branch-&-bound and branch-&-cut approaches, or as a tool to measure the quality of other bounding algorithms. In regard to the form of the given optimization problem, the first step of a relaxation process requires the reformulation of the original problem. The second step comprises the removal or replacement of constraints that are the cause for intractability.

In the following subsections, we review different strategies to derive semidefinite relaxations for the quadratic assignment problem. For reasons of simplicity and comparability, the reviewed relaxations are reduced to their essence. We skip some equalities and any applicable LP and SOCP constraints that are present in more sophisticated versions of these relaxation frameworks. A detailed description of these constraints is given in Section 6.1.

3.1.1. The vector lifting approach

One of the most popular relaxation approaches for quadratic programming problems is based on vector lifting. In the field of optimization, the term “vector lifting” refers

to a specific reformulation approach. Quadratic expressions on a variable $x \in \mathbb{R}^k$ are linearized by lifting the vector variable into the space of $(k+1) \times (k+1)$ matrices.

Consider, for instance, the vectorized formulation of a quadratically constrained quadratic program (QCQP):

$$\begin{aligned} \inf_{x \in \mathbb{R}^k} \quad & x^T H_0 x + h_0^T x \\ \text{s. t.} \quad & x^T H_i x + h_i^T x \leqslant b_i, \quad i = 1, \ldots, m, \end{aligned}$$

where $H_j \in \mathcal{S}^k$, $h_j \in \mathbb{R}^k$ $(0 \leqslant j \leqslant m)$, and $b_i \in \mathbb{R}$ $(1 \leqslant i \leqslant m)$. For a symmetric matrix $\mathcal{Y} \in \mathcal{S}^k$, it is straightforward to show that

$$\mathcal{Y} = xx^T \quad \Longleftrightarrow \quad \mathcal{Y} \succeq xx^T \ \wedge \ \operatorname{diag}(\mathcal{Y}) = \operatorname{diag}(xx^T).$$

It is therefore possible to utilize the Schur complement for the reformulation of QCQP as a non-convex semidefinite programming problem:

$$\begin{aligned} \inf_{x \in \mathbb{R}^k,\ \mathcal{Y} \in \mathcal{S}^k} \quad & \left\langle \begin{bmatrix} 0 & \frac{1}{2} h_0^T \\ \frac{1}{2} h_0 & H_0 \end{bmatrix}, \begin{bmatrix} 1 & x^T \\ x & \mathcal{Y} \end{bmatrix} \right\rangle \\ \text{s. t.} \quad & \left\langle \begin{bmatrix} -b_i & \frac{1}{2} h_i^T \\ \frac{1}{2} h_i & H_i \end{bmatrix}, \begin{bmatrix} 1 & x^T \\ x & \mathcal{Y} \end{bmatrix} \right\rangle \leqslant 0, \quad i = 1, \ldots, m, \\ & \begin{bmatrix} 1 & x^T \\ x & \mathcal{Y} \end{bmatrix} \in \mathcal{S}_+^{k+1}, \quad \operatorname{diag}(\mathcal{Y}) = x \circ x. \end{aligned} \tag{3.1}$$

This problem is often used as a basis for semidefinite relaxations of non-convex quadratic programming problems. The only constraint that has to be dealt with - for the simple reason that it typically causes the intractability of the programming problem - is the quadratic vector equality.

Apparently, the QAP is a special class of QCQP. The usage of the vectorized form of the permutation matrix, $x = \mathrm{vec}(X)$, allows us to formulate (KBQAP) as

$$\begin{aligned} \inf_{x \in \mathbb{R}^{n^2}} \quad & x^T (B \otimes A) x + \langle \mathrm{vec}(C), x \rangle \\ \text{s. t.} \quad & (e \otimes I)^T x = e, \\ & (I \otimes e)^T x = e, \\ & x = x \circ x. \end{aligned} \tag{3.2}$$

The linear equality constraints in (3.2) are the vector versions of $Xe = e$ and $X^T e = e$, which realize the condition $X \in \mathcal{E}^n$. The vector equality $x = x \circ x$ is the quadratic constraint equivalent to $X \in \{0,1\}^{n \times n}$. Together, these constraints implement the requirement $X \in \Pi^n$.

By utilizing the described vector lifting reformulation, we derive a basis of the SDP relaxations discussed in [15, 87, 116] and many other papers:

$$\inf_{x \in \mathbb{R}^{n^2}, Y \in \mathcal{S}^{n^2}} \quad \langle B \otimes A, Y \rangle + \langle \mathrm{vec}(C), x \rangle \tag{3.3a}$$

$$\text{s. t.} \quad \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \in \mathcal{S}_+, \tag{3.3b}$$

$$(e \otimes I)^T x = (I \otimes e)^T x = e, \tag{3.3c}$$

$$\mathrm{diag}(Y) = x. \tag{3.3d}$$

The extension of (3.3c) to $xx^T(e \otimes I) = xx^T(I \otimes e) = xe^T$ leads to the additional vector constraints

$$Y (e \otimes I) = Y (I \otimes e) = xe^T. \tag{3.3e}$$

Finally, for a further restriction of the feasible set, we utilize

$$(E_{\text{off}} \otimes I) \circ \Upsilon = 0 \tag{3.3f}$$

and

$$(I \otimes E_{\text{off}}) \circ \Upsilon = 0. \tag{3.3g}$$

The latter two equations represent $(n^3 - n^2)/2$ additional equality constraints which are called *Gangster* equations.

Problem (3.3) is equivalent to the relaxation QAP_{R_2} that was first introduced in [116]. In order to enhance the comparability with the other presented SDP frameworks, we use a different term to emphasize the basic relaxation approach as well as the level of

additional constraints. Problem (3.3) is based on vector lifting and does not contain additional LP or SOCP constraints. We therefore denote this framework by QAP_{VL_0} or simply VL_0, referring to the Vector Lifting relaxation with level-0 modifications.

For an elaborate discussion of different conic relaxation models based on vector lifting, we refer to [86]. A common strength of the corresponding SDP frameworks is their advantage over other semidefinite programming relaxation strategies in the computation of tight bounds. Their major weakness is the large number of $\mathcal{O}(n^4)$ variables together with the accompanying computational costs.

There are some efforts to encounter the high computational costs of these SDP relaxations. In [87], Rendl and Sotirov improved the computational process by applying a modified version of the bundle method. Other solution approaches designed for this kind of high dimensional SDP frameworks are the low-rank factorization method by Burer and Monteiro [14], and the modified majorized semismooth Newton-CG augmented Lagrangian method by Yang, Sun, and Toh [113]. An entirely different approach to handle the addressed issue is given by Klerk and Sotirov [24], who have shown that many QAP instances allow a reduction of the SDP relaxation by exploiting group symmetries in the underlying problems. Nevertheless, regarding QAP instances of size $n > 30$ with little symmetry, the computational costs of SDP relaxation frameworks based on vector lifting remain to be too high for practical usage.

3.1.2. Matrix lifting

A similar idea to the one presented in Subsection 3.1.1 leads to a very different relaxation approach. In [27], Ding and Wolkowicz introduced a new low-dimensional SDP relaxation framework based on matrix lifting. Instead of lifting the vector $x = \text{vec}(X)$ into $\Upsilon = xx^T$, Ding and Wolkowicz exploited the matrix structure of the Koopmans-Beckmann trace formulation (KBQAP). Their relaxation is based on the fact that for $Y = XBX^T$ and $Z = XB^2X^T$ the matrix

$$\begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} = \begin{bmatrix} I \\ X \\ XB \end{bmatrix} \begin{bmatrix} I & X^T & BX^T \end{bmatrix}$$

is positive semidefinite. Together with the observation that each permutation matrix X and any square matrix G of the same dimension are satisfying the vector equalities

$$\operatorname{diag}(XGX^T) = X\operatorname{diag}(G) \qquad \text{and} \qquad XGX^Te = XGe, \tag{3.4}$$

Ding and Wolkowicz designed the following SDP relaxation:

$$\begin{aligned}
\inf_{X\in\mathcal{D}^n,\ Y,Z\in\mathcal{S}^n} \quad & \langle A, Y\rangle + \langle C, X\rangle && \text{(3.5a)}\\
\text{s. t.} \quad & \begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \in \mathcal{S}_+, && \text{(3.5b)}\\
& \operatorname{diag}(Y) = X\operatorname{diag}(B), \quad \operatorname{diag}(Z) = X\operatorname{diag}(B^2), && \text{(3.5c)}\\
& Ye = XBe, \quad Ze = XB^2e. && \text{(3.5d)}
\end{aligned}$$

This program is the essence of a framework that is called *MSDR*. More precisely, in [27], problem (4.31) is denoted by $MSDR_1$. The final version $MSDR_3$ is given in the projected reformulation of this relaxation and applies additional cuts which are derived from the eigenvalue decomposition of the parameter matrices A and B. We follow our notation from the last subsection and denote problem (3.5) by QAP_{ML_0} or simply by ML_0, referring to the Matrix Lifting technique without further modifications.

3.1.3. The matrix splitting approach

For a special class of QAPs - instances which are associated with Hamming or Manhatten distances - in [73], Mittelmann and Peng pursued the idea of another low-dimensional SDP relaxation framework. The presented bounds not only involve a less expensive computational process, but they are also provably tighter than the ones of the *MSDR* relaxations. In [83] and [84], the matrix splitting approach has been generalized for other classes of the quadratic assignment problem. In this subsection, we are concentrating on the SDP relaxation introduced by Peng, Mittelmann, and Li [83].

If the parameter matrix B is positive semidefinite, the relaxation approach for the *MSDR* model can be reduced to

$$\begin{bmatrix} B & BX^T \\ XB & Y \end{bmatrix} = \begin{bmatrix} B^{\frac{1}{2}} \\ XB^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} B^{\frac{1}{2}} & B^{\frac{1}{2}}X^T \end{bmatrix} \in \mathcal{S}_+^{2n}. \tag{3.6}$$

In general, however, B does not satisfy any definiteness property. The authors of [83] and [84] dealt with this case by splitting B into two positive semidefinite matrices B_1 and B_2.

Definition 3.1. *For a given matrix B, a matrix pair (B_1, B_2) is called a positive semidefinite matrix splitting of B if it satisfies*

$$B = B_1 - B_2, \quad B_1, B_2 \in \mathcal{S}_+. \tag{3.7}$$

For their relaxation frameworks, Peng, Mittelmann, and Li [83] utilized the splitting pair that is derived as the solution of the following trace minimization problem:

$$\begin{aligned} \inf_{B_1, B_2 \in \mathcal{S}_+^n} \quad & \operatorname{tr}(B_1 + B_2) \\ \text{s. t.} \quad & B = B_1 - B_2. \end{aligned} \tag{3.8}$$

As pointed out in [84], the unique solution of this problem can be obtained by using the spectral value decomposition. Let $B = \sum_{i=1}^{n} \lambda_i q_i q_i^T$ denote the eigenvalue decomposition of B, then

$$B_+ := \sum_{i:\ \lambda_i > 0} \lambda_i q_i q_i^T \qquad \text{and} \qquad B_- := \sum_{i:\ \lambda_i < 0} -\lambda_i q_i q_i^T \tag{3.9}$$

define the unique solution pair $(\hat{B}_1, \hat{B}_2) = (B_+, B_-)$ to problem (3.8).

Finally, by utilizing (3.6) and (3.4) for the relaxation of the quadratic constraints $Y_+ = XB_+X^T$ and $Y_- = XB_-X^T$, we derive the Matrix Splitting based framework

referred to as MS_0:

$$\inf_{X\in\mathcal{D}^n,\ Y_+,Y_-\in\mathcal{S}^n} \quad \langle A, Y_+ - Y_-\rangle + \langle C, X\rangle \tag{3.10a}$$

$$\text{s. t.} \quad \begin{bmatrix} B_\diamond & B_\diamond X^T \\ X B_\diamond & Y_\diamond \end{bmatrix} \in \mathcal{S}_+, \qquad \diamond \in \{+,-\}, \tag{3.10b}$$

$$\operatorname{diag}(Y_\diamond) = X \operatorname{diag}(B_\diamond), \qquad \diamond \in \{+,-\}, \tag{3.10c}$$

$$Y_+ e = X B_+ e, \quad Y_- e = X B_- e. \tag{3.10d}$$

The statement $\diamond \in \{+,-\}$ is not meant as a free choice of matrix subscripts, but is used to indicate two instances of constraints of the same structure. In that regard, (3.10c) refers to the two vector equalities $\operatorname{diag}(Y_+) = X \operatorname{diag}(B_+)$ and $\operatorname{diag}(Y_-) = X \operatorname{diag}(B_-)$.

3.1.4. Relaxation of the eigenspace

Based on Ding and Wolkowicz's characterization of the convex hull of the orthogonal similarity set of B [27], Xia gave an SDP formulation for the same convex set. In that context, he proposed the orthogonal bound *OB2*. Klerk, Sotirov, and Truetsch [25] extended this approach by incorporating additional constraints from [83]. Furthermore, they looked into the possibility to reduce the complexity of their framework by exploiting symmetries in the underlying problems.

The "eigenspace" SDP relaxation is, in a certain way, also based on matrix splitting. However, instead of splitting the matrix B just in its positive and negative semidefinite part, this new relaxation makes full use of the eigenvalue decomposition of B:

$$XBX^T = X\left(\sum_{i=1}^n \lambda_i q_i q_i^T\right) X^T = \sum_{i=1}^n \lambda_i X q_i q_i^T X^T = \sum_{i=1}^n \lambda_i Q_i.$$

By applying a similar relaxation approach as in Subsection 3.1.3 to relax the quadratic equality constraints $Q_i = X q_i q_i^T X^T$, we obtain the programming basis of Klerk, Sotirov,

and Truetsch's "eigenspace" SDP relaxation [25]:

$$\inf_{X\in\mathcal{D}^n,\ Q_1,\ldots,Q_n\in\mathcal{S}^n} \quad \sum_{i=1}^n \lambda_i\langle A, Q_i\rangle \ + \ \langle C, X\rangle \tag{3.11a}$$

$$\text{s. t.} \quad \begin{bmatrix} 1 & q_i^T X^T \\ X q_i & Q_i \end{bmatrix} \in \mathcal{S}_+, \qquad i = 1,\ldots,n, \tag{3.11b}$$

$$\operatorname{diag}(Q_i) = X \operatorname{diag}(q_i q_i^T), \qquad i = 1,\ldots,n, \tag{3.11c}$$

$$Q_i e = X q_i q_i^T e, \qquad i = 1,\ldots,n, \tag{3.11d}$$

$$\sum_{i=1}^n Q_i = I. \tag{3.11e}$$

The last identity is derived from the orthogonality of the eigenvectors $\{q_i\}$ giving $\sum_{i=1}^n X q_i q_i^T X^T = XIX^T = I$. For the identification in the following sections, we comply with the introduced notation and denote this relaxation program by ES_0, referring to the "eigenspace" SDP relaxation without modifications and additional LP or SOCP constraints.

3.2. Comparison and other relaxation properties

Regarding the number of variables and equality constraints, it is evident that the mentioned relaxation frameworks involve strongly different computational efforts. In respect of the tightness of the computed bounds, we expect the larger sized frameworks, VL_0 and ES_0, to dominate the lower sized ones. Our numerical tests meet this expectation.

For an easier interpretation of the relaxation quality, the corresponding results are presented in form of relative gaps

$$R_{\text{gap}} := 1 - \frac{\text{Lower bound computed via relaxation}}{\text{Best known upper bound or optimal value}}. \tag{3.12}$$

The QAPs used for the numerical examples in Table 3.1 are chosen randomly out of the quadratic assignment problem library [18]. Here, we limited the selection to QAP instances with dimension n up to 20. The individual problem sizes are incorporated into the names. For more information on the naming scheme and the individual applications, see [18].

Table 3.1.: Selected bounds for comparison of basic relaxation concepts [R_{gap} in (%)]

Problem	VL_0	ML_0	MS_0	ES_0
Chr12a	**150.68**	231.75	375.32	200.36
Esc16b	**5.48**	10.15	17.34	**5.48**
Had14	**0.56**	7.29	5.07	2.00
LiPa20a	**0.72**	1.76	5.13	1.34
Nug12	**8.42**	21.24	20.26	10.50
Scr20	**24.26**	80.07	60.02	30.26
Tai17a	**10.27**	15.71	29.06	13.18

The presented bounds demonstrate that there is no ordering between the quality of the relaxation programs ML_0 and MS_0.[1] On the other hand, we can show that the ordering of the other relaxation bounds holds valid independently of the considered problem instance.

Theorem 3.2. *For a QAP instance (A, B, C), denote by $\hat{\varrho}_{\text{VL}_0}$, $\hat{\varrho}_{\text{ML}_0}$, $\hat{\varrho}_{\text{MS}_0}$, and $\hat{\varrho}_{\text{ES}_0}$ the optimal objective values to the relaxations* (3.3), (3.5), (3.10), *and* (3.11), *respectively. These values satisfy the relation*

$$\hat{\varrho}_{\text{VL}_0} \quad \geqslant \quad \hat{\varrho}_{\text{ES}_0} \quad \geqslant \quad \begin{cases} \hat{\varrho}_{\text{ML}_0} \\ \hat{\varrho}_{\text{MS}_0} \end{cases} . \tag{3.13}$$

Proof. Denote by $B = \sum_{i=1}^{n} \lambda_i q_i q_i^T$ the eigenvalue decomposition of B. Moreover, for $1 \leqslant i \leqslant n$, define the matrices $W_i := q_i \otimes I$ as well as their horizontal concatenation $W := [W_1, \ldots, W_n]$. Let $(\hat{x}, \hat{Y})$ denote a solution vector to problem (3.3), which thereby gives the identity

$$\hat{\varrho}_{\text{VL}_0} = \langle B \otimes A, \hat{Y} \rangle + \langle \text{vec}(C), \hat{x} \rangle.$$

For the proof of the inequality $\hat{\varrho}_{\text{VL}_0} \geqslant \hat{\varrho}_{\text{ES}_0}$, we show that

$$(X, Q_1, \ldots, Q_n) := \left(\text{mat}(\hat{x}), W_1^T \hat{Y} W_1, \ldots, W_n^T \hat{Y} W_n \right)$$

[1] This observation does not support [73, Theorem 4.1].

defines a feasible point to problem (3.11) with objective value $\hat{\varrho}_{\mathrm{VL0}}$.

Due to the construction of W from the eigenvectors of B,

$$W^T(B \otimes A)W = \sum_{i=1}^{n} W^T \left((\lambda_i q_i q_i^T) \otimes A\right) W = \overset{*}{\mathrm{diag}}([\lambda_1, \ldots, \lambda_n]) \otimes A$$

is a block-diagonal matrix. Together with the orthogonality of W, we derive

$$\langle B \otimes A, \hat{Y} \rangle = \langle W^T(B \otimes A)W, W^T \hat{Y} W \rangle = \sum_{i=1}^{n} \lambda_i \langle A, W_i^T \hat{Y} W_i \rangle = \sum_{i=1}^{n} \lambda_i \langle A, Q_i \rangle,$$

which validates identical objective values.

In the introduction of relaxation (3.3), it has already been pointed out that any feasible vector variable x satisfies $\mathrm{mat}(x) \in \mathcal{D}^n$. Apparently, $X \in \mathcal{D}^n$ is feasible in problem (3.11). It remains to show that the matrices $\{Q_i\}$ satisfy all other constraints of ES_0. The validity of the semidefiniteness conditions in (3.11b) is due to (3.3b) and the implication

$$\hat{Y} \succeq \hat{x}\hat{x}^T \quad \Longrightarrow \quad Q_i = W_i^T \hat{Y} W_i \succeq W_i^T \hat{x}\hat{x}^T W_i = X q_i q_i^T X^T \qquad (1 \leqslant i \leqslant n).$$

On the other hand, (3.3d) and (3.3f) necessitate the validity of the equality constraints on the diagonal elements of $\{Q_i\}$:

$$\begin{aligned} \forall_{i,j}\ 1 \leqslant i, j \leqslant n: \quad e_j W_i^T \hat{Y} W_i e_j &= (q_i \otimes e_j)^T \hat{Y} (q_i \otimes e_j) \\ &= \langle \hat{Y}, (q_i q_i^T) \otimes (e_j e_j^T) \rangle \\ &= \langle \hat{Y}, (q_i q_i^T)_{\mathrm{diag}} \otimes (e_j e_j^T) \rangle \qquad \text{(by 3.3f)} \\ &= e_j^T X \,\mathrm{diag}(q_i q_i^T) \qquad \text{(by 3.3d)}. \end{aligned}$$

Similarly, (3.3e) implies

$$Q_i e = W_i^T \hat{Y}(q_i \otimes e) = W_i^T \left(\hat{Y}(I \otimes e)\right) q_i = W_i^T \left(\hat{x} e^T\right) q_i = X q_i q_i^T e,$$

which requires the compliance of (3.11d). Finally,

$$\begin{aligned}\sum_{i=1}^{n} W_i^T \hat{\Upsilon} W_i &= \sum_{i=1}^{n} (e \otimes I)^T \left(\hat{\Upsilon} \circ (q_i q_i^T \otimes E) \right) (e \otimes I) \\ &= (e \otimes I)^T \left(\hat{\Upsilon} \circ (I \otimes E) \right) (e \otimes I) \qquad \left(\textstyle\sum_{i=1}^{n} q_i q_i^T = I\right) \\ &= (e \otimes I)^T \operatorname{diag}^*(\hat{x}) \, (e \otimes I) \qquad \text{(by 3.3g and 3.3d)} \\ &= I \qquad \text{(by 3.3c)}\end{aligned}$$

verifies the validity of (3.11e) and finishes the proof for the first inequality.

The argument for the superiority of $\hat{\varrho}_{\mathrm{ES}_0}$ compared to $\hat{\varrho}_{\mathrm{ML}_0}$ and $\hat{\varrho}_{\mathrm{MS}_0}$ follows a very similar approach. For a solution vector $(\hat{X}, \hat{Q}_1, \ldots, \hat{Q}_n)$ to problem (3.11), we show that

$$(X, Y, Z) := \Big(\hat{X}, \; \sum_{i=1}^{n} \lambda_i \hat{Q}_i, \; \sum_{i=1}^{n} \lambda_i^2 \hat{Q}_i \Big)$$

and

$$(X, Y_+, Y_-) := \Big(\hat{X}, \; \sum_{i:\, \lambda_i > 0} \lambda_i \hat{Q}_i, \; -\sum_{i:\, \lambda_i < 0} \lambda_i \hat{Q}_i \Big)$$

define feasible points to the problems (3.5) and (3.10), respectively.

Obviously, both points accompany the same objective value $\hat{\varrho}_{\mathrm{ES}_0}$. Moreover, by construction, (3.11c) implies the compliance of (3.5c) and (3.10c). This relation is also valid for the constraints (3.11d), (3.5d), and (3.10d). By the same token,

$$Y_+ = \sum_{i:\, \lambda_i > 0} \lambda_i \hat{Q}_i \;\succeq\; \sum_{i:\, \lambda_i > 0} \lambda_i \hat{X} q_i q_i^T \hat{X}^T = X B_+ X^T$$

and

$$Y_- = \sum_{i:\, \lambda_i < 0} \lambda_i \hat{Q}_i \;\succeq\; \sum_{i:\, \lambda_i < 0} \lambda_i \hat{X} q_i q_i^T \hat{X}^T = X B_- X^T$$

verify the compliance with (3.10b). Finally, the nonnegative definiteness of

$$\begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} q_i & 0_{(n,n)} \\ 0 & I \\ 0 & \lambda_i I \end{bmatrix} \underbrace{\begin{bmatrix} 1 & q_i^T \hat{X}^T \\ \hat{X} q_i & \hat{Q}_i \end{bmatrix}}_{\succeq 0} \begin{bmatrix} q_i & 0_{(n,n)} \\ 0 & I \\ 0 & \lambda_i I \end{bmatrix}^T$$

completes the proof. □

It should be noted that the first inequality on the left-hand side of (3.13) has already been proved in [25, Theorem 2.1]. We believe that the understanding of the connection between the feasible set to problem (3.3) and subsets of the feasible sets to the other presented SDP relaxations is essential for the comprehension of results given in Subsection 4.3 and 6.1.3. For this reason, though the corresponding inequality has already been shown by Klerk, Sotirov, and Truetsch, we still gave the exploited part of their argument adapted to our notation.

All four SDP relaxations have the following favorable property in common.

Lemma 3.3. *For a given QAP instance, let* $(\hat{x}, \hat{Y})$ *denote a solution vector to relaxation* (3.3) *and define* $\hat{X} := \mathrm{mat}(\hat{x})$*. If* $\hat{X} \in \Pi$*, then* $\hat{X}$ *describes an optimal assignment for the actual QAP. The corresponding objective values are identical. Similarly, this statement applies to the matrices* $\hat{X}$ *of solution vectors to the relaxations* (3.5)*,* (3.10)*, and* (3.11)*, respectively.*

Proof. As an example, consider relaxation (3.5). Since $\hat{X} \in \Pi$, it is

$$\hat{X}\hat{X}^T = I \qquad \text{and} \qquad \mathrm{diag}(\hat{X}B^2\hat{X}^T) = \hat{X}\,\mathrm{diag}(B^2).$$

By (3.5c), we therefore have

$$\mathrm{diag}\left(\begin{bmatrix} I & \hat{Y} \\ \hat{Y} & \hat{Z} \end{bmatrix}\right) = \mathrm{diag}\left(\begin{bmatrix} \hat{X} \\ \hat{X}B \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{X}B \end{bmatrix}^T\right).$$

From the Schur complement condition, it is clear that the semidefinite constraint in (3.5b) requires the quadratic residual

$$R := \begin{bmatrix} I & \hat{Y} \\ \hat{Y} & \hat{Z} \end{bmatrix} - \begin{bmatrix} \hat{X} \\ \hat{X}B \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{X}B \end{bmatrix}^T$$

to be positive semidefinite. By $\mathrm{diag}(R) \equiv 0$ and thus $S \equiv 0$, the statement of Lemma 3.3 follows immediately.

The arguments for the other SDP relaxations follow the same procedure. □

Chapter 4.

Reformulation Approaches

In the context of relaxation programs, there are two kinds of reformulation approaches. The first involves modifications of the original programming problem, here the QAP given in Koopmans-Beckmann trace formulation (KBQAP). Major modifications, as for instance the reformulation via vector lifting or the splitting of parameter matrices, specify the respective relaxation technique. Adjustments on a smaller scale do not affect the actual relaxation approach but still influence the feasible sets described by the respective programming instances. The second kind of reformulations concerns the relaxation program itself and is therefore not related to variations in the bounding quality. On the other hand, modifications of this kind can exert a significant influence on the applied solving methods.

In the following sections, we describe ways to improve the quality of the relaxation frameworks via reformulation techniques and point out different constraint substitutions to design more efficient versions of these frameworks.

4.1. Reduction via QAP reformulation

4.1.1. QAP reformulations

The first applications of reformulation strategies for the quadratic assignment problem date back nearly 40 years. The proposed techniques in the works of Burkard and Stratmann [19], Roucairol [91], and Edwards [31] were taken up by many other scientists who worked on similar topics, see [3, 20, 35, 55] and the references therein. In earlier research papers, these techniques are noted as reduction or decomposition schemes. The

background is the same for all proposed reformulations strategies. One tries to exploit the inherent degree of freedom in reformulating the actual problem instance without affecting the optimal objective value or the accompanying optimal decision variables. This freedom is used to construct problem instances with the same solution sets but more beneficial properties for the considered linearization technique. In later works, it has been shown that the benefit of reformulations is not limited to relaxations which are based on primal or dual linearizations of the QAP. Finke, Burkard, and Rendl [34] as well as Hadley, Rendl, and Wolkowicz [44], for example, demonstrated the applicability of QAP reformulations to eigenvalue based bounding techniques.

In this subsection, we transfer the conceptional basis of these reduction schemes for the application in the presented SDP relaxations. For this purpose, we first recap the basic reformulation procedure and then explain which parameter adjustments can be performed independently and which are actually affecting the discussed SDP relaxations. This is followed by a thorough analysis of different strategies to obtain appropriate reformulation parameters.

Except for the vector lifting relaxation approach, all discussed SDP frameworks require the form (KBQAP). We therefore limit our consideration on reformulation techniques which preserve this form. In accordance to [84], we describe a reformulation by an update of a diagonal matrix and a so-called sum-matrix.

Definition 4.1. *A matrix $M \in \mathcal{M}^n$ is called a sum-matrix if M is representable as*

$$M = ve^T + ew^T \tag{4.1}$$

for some $v, w \in \mathbb{R}^n$.

Definition 4.2. *A reformulation of a QAP in Koopmans-Beckmann form is another QAP with parameter vectors $d_A, d_B, v_A, v_B, w_A, w_B \in \mathbb{R}^n$, where*

$$\acute{A} := A + \operatorname{diag}^*(d_A) + v_A e^T + e w_A^T, \tag{4.2a}$$

$$\acute{B} := B + \operatorname{diag}^*(d_B) + v_B e^T + e w_B^T, \tag{4.2b}$$

$$\acute{C} := C - \operatorname{diag}(\acute{A}) d_B^T - d_A \operatorname{diag}(B)^T - \acute{A}^T e v_B^T - \acute{A} e w_B^T - v_A e^T B - w_A e^T B^T, \tag{4.2c}$$

define substitutes for the original coefficient matrices.

It is straightforward to show (c.f. [34]) that for each choice of parameter vectors d_A, d_B, v_A, v_B, w_A, w_B and every permutation matrix $X \in \Pi^n$:

$$\operatorname{tr}(AXBX^T + CX^T) = \operatorname{tr}(\acute{A}X\acute{B}X^T + \acute{C}X^T). \tag{4.3}$$

In the symmetric case, it is $v_A = w_A$ and $v_B = w_B$, which - in consideration of the initial assumptions on the data matrices - reduces (4.2c) to

$$\acute{C} := C - d_A d_B^T - 2\acute{A}ev_B^T - 2v_A e^T B. \tag{4.4}$$

We aim to find parameter vectors for reformulation instances which turn out to be particularly beneficial for semidefinite programming relaxations. The respective relaxation program shall thus deliver stronger bounds when applied to the reformulated QAP instance $(\acute{A}, \acute{B}, \acute{C})$ instead of the original formulation (A, B, C). For this purpose, we first collect some useful properties which are related to the SDP relaxations discussed in Section 3.1. Our first result concerns the framework based on the vector lifting approach.

Lemma 4.3. *Relaxation* (3.3) *delivers the same optimal objective value for every QAP reformulation* $(\acute{A}, \acute{B}, \acute{C})$ *of the same problem instance* (A, B, C).

Proof. Denote by (x, Y) an arbitrary feasible point to the considered problem instance of relaxation (3.3). Obviously, the feasible set to this problem solely depends on the dimension n and not on the specific coefficient matrices. It is therefore sufficient to prove the equality

$$\langle B \otimes A, Y\rangle + \langle \operatorname{vec}(C), x\rangle = \langle \acute{B} \otimes \acute{A}, Y\rangle + \langle \operatorname{vec}(\acute{C}), x\rangle \tag{4.5}$$

for all valid $(\acute{A}, \acute{B}, \acute{C})$.

By definition,

$$\begin{aligned}
\Big\langle (\acute{B} - B) \otimes \acute{A}, Y \Big\rangle &= \Big\langle \big(\operatorname{diag}^*(d_B) + v_B e^T + e w_B^T \big) \otimes \acute{A}, Y \Big\rangle \\
&= \Big\langle \operatorname{diag}^*(d_B) \otimes \acute{A}, Y \Big\rangle + \Big\langle \big(e v_B^T\big) \otimes \acute{A}^T + \big(e w_B^T\big) \otimes \acute{A}, Y \Big\rangle \\
&= \Big\langle \operatorname{diag}^*(d_B) \otimes \acute{A}, Y \Big\rangle + \Big\langle \big(e \otimes I\big)\big(v_B^T \otimes \acute{A}^T + w_B^T \otimes \acute{A}\big), Y \Big\rangle \\
&= \Big\langle \operatorname{diag}^*(d_B) \otimes \acute{A}, Y \Big\rangle + \Big\langle v_B^T \otimes \acute{A}^T + w_B^T \otimes \acute{A}, Y\big(e \otimes I\big) \Big\rangle .
\end{aligned}$$

Together with the constraint equalities (3.3e), (3.3d), and (3.3g), we derive

$$\begin{aligned}\Big\langle (\acute{B} - B) \otimes \acute{A}, \Upsilon \Big\rangle &= \Big\langle d_B \otimes \operatorname{diag}(\acute{A}), \operatorname{diag}(\Upsilon) \Big\rangle + \Big\langle v_B^T \otimes \acute{A}^T + w_B^T \otimes \acute{A}, x e^T \Big\rangle \\ &= \Big\langle \operatorname{vec}(\operatorname{diag}(\acute{A}) d_B^T), x \Big\rangle + \Big\langle \operatorname{vec}(\acute{A}^T e v_B^T + \acute{A} e w_B^T), x \Big\rangle .\end{aligned}$$

By the same argument, it can be shown that

$$\Big\langle B \otimes (\acute{A} - A), \Upsilon \Big\rangle = \big\langle \operatorname{vec}(d_A \operatorname{diag}(B)^T), x \big\rangle + \big\langle \operatorname{vec}(v_A e^T B + w_A e^T B^T), x \big\rangle .$$

Adding the latter two identities gives

$$\langle (\acute{B} \otimes \acute{A}) - (B \otimes A), \Upsilon \rangle = \langle (\acute{B} - B) \otimes \acute{A}, \Upsilon \rangle + \langle B \otimes (\acute{A} - A), \Upsilon \rangle = \langle \operatorname{vec}(C - \acute{C}), x \rangle.$$

This validates (4.5) and finishes the proof. □

Lemma 4.4. *For a given problem (A, B, C), denote by $\{(\acute{A}, B, \acute{C})\}$ the set of valid reformulations with fixed B. The choice of a particular instance from this set does not affect the bounds computed via problem* (3.5)*,* (3.10)*, or* (3.11)*, respectively.*

Proof. The following proof is similar to that of Lemma 4.3. The feasible sets of the problems (3.5), (3.10), and (3.11) are independent of the coefficient matrices A and C. Thus, in case of fixed B, it is sufficient to validate the equivalence of the corresponding objective values.

As an example, consider problem (3.10) and let $(X, Y_+, Y_-) \in \mathcal{D}^n \times \mathcal{S}^n \times \mathcal{S}^n$ denote an arbitrary feasible point to the given problem instance. By Definition 4.2 and by the compliance of (X, Y_+, Y_-) with the constraints (3.10c), (3.10d), we conclude

$$\begin{aligned}\big\langle \acute{A} - A, Y_+ - Y_- \big\rangle &= \big\langle \operatorname{diag}^*(d_A) + v_A e^T + e w_A^T, Y_+ - Y_- \big\rangle \\ &= \big\langle d_A, \operatorname{diag}(Y_+ - Y_-) \big\rangle + \big\langle v_A + w_A, (Y_+ - Y_-) e \big\rangle \\ &= \big\langle d_A, X \operatorname{diag}(B_+ - B_-) \big\rangle + \big\langle v_A + w_A, X(B_+ - B_-) e \big\rangle \\ &= \big\langle d_A \operatorname{diag}(B)^T, X \big\rangle + \big\langle (v_A + w_A) e^T B, X \big\rangle \\ &= \big\langle C - \acute{C}, X \big\rangle,\end{aligned}$$

which finishes the argument for problem (3.10).

In the same way, the equality constraints (3.5c), (3.5d) and (3.11c), (3.11d) necessitate the equivalence of the objective values for relaxation (3.5) and (3.11), respectively. □

As an immediate consequence of these lemmas, we see that solely updates on the coefficient matrix B have an effect on the corresponding relaxation bounds. The interesting reformulation parameters reduce to $d_B, v_B \in \mathbb{R}^n$, where $w_B = v_B$ is implicit due to the symmetry assumption. The following result will help us to determine a sensible choice for the vector v_B.

Lemma 4.5. *For a given problem instance (A, B, C), consider the QAP reformulation $(\acute{A}, \acute{B}, \acute{C})$, where*

$$\acute{A} = A, \quad \acute{B} = (I - \tfrac{1}{n}E)B(I - \tfrac{1}{n}E) \quad \textit{and} \quad \acute{C} = C + \tfrac{2}{n}AEB(I - \tfrac{1}{2n}E). \tag{4.6}$$

The respective optimal objective values $\hat{\varrho}_{\mathrm{ML}_0}(A, B, C)$ and $\hat{\varrho}_{\mathrm{ML}_0}(\acute{A}, \acute{B}, \acute{C})$ to problem (3.5) satisfy the relation

$$\hat{\varrho}_{\mathrm{ML}_0}(\acute{A}, \acute{B}, \acute{C}) \geqslant \hat{\varrho}_{\mathrm{ML}_0}(A, B, C). \tag{4.7}$$

Proof. For all $X \in \Pi^n$, we have

$$\begin{aligned}
\langle \acute{A}X\acute{B} + \acute{C}, X\rangle &= \langle AX(I - \tfrac{1}{n}E)B(I - \tfrac{1}{n}E) + \acute{C}, X\rangle \\
&= \langle AXB(I - \tfrac{1}{n}E) - \tfrac{1}{n}AEB(I - \tfrac{1}{n}E) + \acute{C}, X\rangle \\
&= \langle AXB(I - \tfrac{1}{n}E) + \tfrac{1}{n}AEB + C, X\rangle \\
&= \langle AXB + C, X\rangle.
\end{aligned}$$

It is therefore evident that the problem instance $(\acute{A}, \acute{B}, \acute{C})$ defined in (4.6) states a reformulation of the original problem (A, B, C).

Denote by $ML_0(A, B, C)$ and $ML_0(\acute{A}, \acute{B}, \acute{C})$ the respective problem instances of relaxation (3.5), the former applied to the original QAP and the latter applied to its reformulation. Define the projection matrix $P := I - \frac{1}{n}E$ and let $(\acute{X}, \acute{Y}, \acute{Z})$ be a feasible point to $ML_0(\acute{A}, \acute{B}, \acute{C})$. By applying the Schur complement, it can be shown that (3.5b) and

$$\begin{bmatrix} I - \acute{X}\acute{X}^T & \acute{Y} - \acute{X}\acute{B}\acute{X}^T \\ \acute{Y} - \acute{X}\acute{B}\acute{X}^T & \acute{Z} - \acute{X}\acute{B}^2\acute{X}^T \end{bmatrix} \in \mathcal{S}_+^{2n} \tag{4.8}$$

are equivalent, i.e. they result in the same feasible set of the variables $\acute{Y}$ and $\acute{Z}$. Let R be a symmetric positive semidefinite matrix satisfying

$$R \geq \acute{X}(PB^2P - \acute{B}^2)\acute{X}^T, \quad Re = 0, \quad \operatorname{diag}(R) = \acute{X}\operatorname{diag}(PB^2P - \acute{B}^2).$$

By Lemma 2.15 and the positive semidefiniteness of $PB^2P - \acute{B}^2 = PB(I-P)BP$, it is apparent that such a matrix exists for each $\acute{X} \in \mathcal{D}^n$. Let further

$$(X, Y, Z) := \left(\acute{X},\ \acute{Y} + \acute{X}(B - \acute{B})\acute{X}^T,\ \acute{Z} + R + \acute{X}(B^2 - PB^2P)\acute{X}^T\right).$$

The semidefiniteness condition in (4.8) together with the equations

$$Y - XBX^T = \acute{Y} + X(B - \acute{B})X^T - XBX^T = \acute{Y} - X\acute{B}X^T$$

and

$$Z - XB^2X^T = \acute{Z} + R - XPB^2PX^T \geq \acute{Z} - X\acute{B}^2X^T$$

imply

$$\begin{bmatrix} I - XX^T & Y - XBX^T \\ Y - XBX^T & Z - XB^2X^T \end{bmatrix} \in \mathcal{S}_+^{2n}.$$

Hence, the constructed point (X, Y, Z) complies with constraint (3.3b) of $ML_0(A, B, C)$. It is straightforward to check that the matrix triple (X, Y, Z) satisfies the corresponding equality constraints in (3.5c) and (3.5d), thus states a feasible point to $ML_0(A, B, C)$. Finally, the identity

$$\begin{aligned} \langle \acute{A}, \acute{Y} \rangle + \langle \acute{C}, \acute{X} \rangle &= \langle A, Y - X(B - \acute{B})X^T \rangle + \langle \acute{C}, X \rangle \\ &= \langle A, Y - \tfrac{1}{n}X(EB + PBE)X^T \rangle + \langle \acute{C}, X \rangle \\ &= \langle A, Y \rangle - \tfrac{1}{n}\langle A, EBX^T + XPBE \rangle + \langle \acute{C}, X \rangle \\ &= \langle A, Y \rangle + \langle C, X \rangle \end{aligned}$$

proves that, for every feasible point $(\acute{X}, \acute{Y}, \acute{Z})$ to $ML_0(\acute{A}, \acute{B}, \acute{C})$, there exists a feasible point (X, Y, Z) to $ML_0(A, B, C)$ with the same objective value. □

Let the reformulation vector d_B for the diagonal elements of $\acute{B}$ be fixed, and consider the specific choice for v_B defined by

$$v_B := \frac{\langle B, E\rangle + \langle d_B, e\rangle}{2n^2} e - \frac{1}{n}\left(Be + d_B\right). \tag{4.9}$$

This vector satisfies the identity

$$(I - \tfrac{1}{n}E)\left(B + \operatorname{diag}^*(d_B)\right)(I - \tfrac{1}{n}E) = B + \operatorname{diag}^*(d_B) + v_B e^T + e v_B^T.$$

Lemma 4.5 therefore implies that the choice of the specific reformulation vector v_B defined in (4.9) is optimal for any QAP instance that shall be used as input to relaxation (3.5).

Though the previous statement is not necessarily valid for ES_0 and MS_0, numerical tests suggest the application of the same reformulation vector v_B. With the following result, we reinforce the chosen matrix splitting approach defined in (3.9) and confirm the formula for v_B.

Lemma 4.6. *Let $B \in \mathcal{S}^n$ be given and consider the minimization problem*

$$\begin{aligned} \inf_{v_B \in \mathbb{R}^n,\ \mathsf{B}_1, \mathsf{B}_2 \in \mathcal{S}^n_+} \quad & w_1 |\!|\!| \mathsf{B}_1 |\!|\!| + w_2 |\!|\!| \mathsf{B}_2 |\!|\!| \\ \text{s. t.} \quad & B + \mathsf{v}_B e^T + e \mathsf{v}_B^T = \mathsf{B}_1 - \mathsf{B}_2, \end{aligned} \tag{4.10}$$

where $|\!|\!| \cdot |\!|\!|$ denotes some unitarily invariant matrix norm, and w_1, w_2 are arbitrary real positive values. Furthermore, define the projected matrix $\acute{B} := (I - \frac{1}{n}E)B(I - \frac{1}{n}E)$ with eigenvalue decomposition $\acute{B} = \sum_{i=1}^n \acute{\lambda}_i \acute{q}_i \acute{q}_i^T$. Then,

$$\acute{v}_B = \frac{\langle B, E\rangle}{2n^2} e - \frac{1}{n} Be, \quad \acute{B}_+ = \sum_{i:\ \lambda_i > 0} \acute{\lambda}_i \acute{q}_i \acute{q}_i^T \quad \text{and} \quad \acute{B}_- = \sum_{i:\ \lambda_i < 0} -\acute{\lambda}_i \acute{q}_i \acute{q}_i^T \tag{4.11}$$

determine a solution vector $(\hat{\mathsf{v}}_B, \hat{\mathsf{B}}_1, \hat{\mathsf{B}}_2) = (\acute{v}_B, \acute{B}_+, \acute{B}_-)$ to problem (4.10). If the considered matrix norms are strictly monotone, this solution is unique.

Proof. Define the projection matrix $P := I - \frac{1}{n}E$. In the first part of the proof, we will show that, for any feasible point $(\mathsf{v}_B, \mathsf{B}_1, \mathsf{B}_2)$, there exists another feasible point $(\acute{v}_B, P\mathsf{B}_1 P, P\mathsf{B}_2 P)$ associated with an objective value not greater than the one accompanied by $(\mathsf{v}_B, \mathsf{B}_1, \mathsf{B}_2)$. The positive semidefiniteness of $P\mathsf{B}_1 P$ and $P\mathsf{B}_2 P$ is evidently satisfied. Moreover

$$B + \acute{v}_B e^T + e \acute{v}_B^T = PBP = P(B + \mathsf{v}_B e^T + e \mathsf{v}_B^T)P = P(\mathsf{B}_1 - \mathsf{B}_2)P$$

validates the compliance with the equality constraint of problem (4.10). Since P is an orthogonal projection matrix and thereby $\|\!|P|\!\|_2 \leqslant 1$, it follows $\sigma^{\downarrow}(PB_iP) \leqslant \sigma^{\downarrow}(B_i)$ for $i = 1, 2$. By Theorem 2.3, this necessarily implies $\|\!|PB_iP|\!\| \leqslant \|\!|B_i|\!\|$ independent of the choice of the unitarily invariant matrix norm. Taken together, these observations prove the existence of an optimal point $(\acute{v}_B, \hat{B}_1, \hat{B}_2)$ with $\acute{v}_B$ being obtained from (4.11). Strictly monotone unitarily invariant matrix norms satisfy

$$B_iP \neq B_i \implies \|\!|PB_iP|\!\| < \|\!|B_i|\!\| \qquad \text{for} \quad i = 1, 2,$$

which, in this context, proves the uniqueness of $\hat{v}_B$.

For each solution vector $(\acute{v}_B, \hat{B}_1, \hat{B}_2)$ to problem (4.10) for which $\acute{v}_B$ satisfies the definition in (4.11), there exists a feasible point $(G, H) = (0_{(n,n)}, \hat{B}_1)$ to the semidefinite programming problem

$$\begin{aligned} \inf_{G,H \in \mathcal{S}^n} \quad & w_1\|\!|H|\!\| + w_2\|\!|H - PBP|\!\| \\ \text{s. t.} \quad & \begin{bmatrix} H & G \\ G & H - PBP \end{bmatrix} \in \mathcal{S}^{2n}_+, \\ & q_i^T G q_i = 0 \qquad \text{for} \quad i \in \{1, \ldots, n\}. \end{aligned}$$

Moreover, the equality constraints in problem (4.10) imply $\hat{B}_2 = \hat{B}_1 - PBP$ and thereby identical objective values. Conversely, any optimal point $(\hat{G}, \hat{H})$ to the above problem that corresponds to a feasible point $(v_B, B_1, B_2) = (\acute{v}_B, \hat{H}, \hat{H} - PBP)$ to problem (4.10) necessarily describes a solution to both. The remaining assertions follow by Lemma 2.16. □

In order to obtain good quality lower bounds via problem (3.10), it is of major importance that the semidefiniteness conditions in (3.10b) approximate the quadratic part of the QAP very well. In this context, it is beneficial to utilize positive semidefinite splitting schemes that involve small traces of the corresponding splitting parts $\acute{B}_+, \acute{B}_-$. The rough correlation between the tightness of the respective relaxation instance and the traces of $\acute{B}_+, \acute{B}_-$ is also apparent from the discussion about non-redundant matrix splitting schemes in [84]. Furthermore, to a certain extent, this statement also applies to the "eigenspace" SDP relaxation as well as the frameworks based on the matrix lifting approach. The triple $(\acute{v}_B, \acute{B}_+, \acute{B}_-)$ complies with v_B given in (4.9) and the matrix splitting definition in (3.9). It states a minimizer not only for the traces of the positive semidefinite

matrix splitting parts but for all unitarily invariant norms. This statement holds true for different weightings and even for arbitrary combinations of different unitarily invariant matrix norms applied individually to the respective positive semidefinite splitting part. For fixed reformulation parameters $(d_B, v_B) = (0, \acute{v}_B)$, the matrix pair $(\acute{B}_+, \acute{B}_-)$ states a solution to the corresponding rank minimization problem discussed in [84, Corollary 1]. It is, furthermore, the only feasible positive semidefinite splitting with orthogonal column spaces. This accompanies other favorable properties, for instance, the fact that $\acute{B}_+$ and $\acute{B}_-$ are simultaneously diagonalizable.

If the reformulation vector v_B is computed via (4.9), then e lies in the kernel of the parameter matrix $\acute{B}$, i.e. $\acute{B}e \equiv 0$. A closer inspection of the related QAP instances $(\acute{A}, \acute{B}, \acute{C})$ reveals that a further reformulation by adding some offset $\alpha \in \mathbb{R}$ - which refers to the reformulated instance $(\grave{A}, \grave{B}, \grave{C}) = (\acute{A}, \acute{B} + \alpha E, \acute{C} - \alpha \acute{A} E)$ - has no effect on the accompanying optimal objective values. This is true for the four reviewed SDP relaxations (3.3), (3.5), (3.10), (3.11), and remains true for all frameworks that will be presented in the upcoming sections. In this regard, every vector $v_B \in \{\alpha e - \frac{1}{n}(Be + d_B) \mid \alpha \in \mathbb{R}\}$ is as good as the choice (4.9). Nevertheless, as already pointed out, the utilization of the particular vector determined by the formula in (4.9) entails certain advantages. A particularly useful property is the minimal rank of the splitting parts which can be exploited for a reduction of the corresponding semidefinite constraints.

With the knowledge that the parameters d_A and v_A have no influence on the relaxation quality and the attainment of a well reasoned formula for v_B, this leaves solely the problem of finding appropriate choices for the reformulation parameter d_B. The task of finding the optimal vector d_B is significantly more difficult than for the parameter v_B. Actually, there are several choices for d_B which are differently suitable for different QAP instances. Keeping this in mind, we want to complete our investigation of QAP reformulations by discussing two possible choices for d_B with different beneficial properties.

The first one is given explicitly by the formula

$$d_B = \frac{2}{n-2} Be - \frac{\langle B, E \rangle}{(n-1)(n-2)} e. \tag{4.12}$$

Together with (4.9), the derived reformulation follows the reduction scheme already used in [34] for the bounding technique *EVB1*. The given parameters minimize the Frobenius norm of the reformulated data matrix $\acute{B}$. By $\operatorname{tr}(\acute{B}^2) = |||\acute{B}|||_{\mathrm{F}}^2$, it is apparent that this involves a minimization of the trace of the matrix variable Z utilized in problem (3.5).

The latter observation explains why (4.12) gives a sensible choice for the corresponding instances of QAP_{ML}.

The second choice for $d_{\scriptscriptstyle B}$ can be derived from the solution to the following semidefinite programming problem

$$\begin{aligned} \inf_{d_{\scriptscriptstyle B}\in\mathbb{R}^n,\ \mathcal{B}_1,\mathcal{B}_2\in\mathcal{S}^n_+} \quad & w_1\,\mathrm{tr}(\mathcal{B}_1) + w_2\,\mathrm{tr}(\mathcal{B}_2) \\ \text{s. t.} \quad & (I - \tfrac{1}{n}E)\left(B + \mathrm{diag}^*(d_{\scriptscriptstyle B})\right)(I - \tfrac{1}{n}E) = \mathcal{B}_1 - \mathcal{B}_2, \end{aligned} \tag{4.13}$$

where the coefficients w_1 and w_2 in the objective function are used to induce a suitable weighting of the respective positive semidefinite splitting parts. Let $(\hat{d}_{\scriptscriptstyle B}, \hat{\mathcal{B}}_1, \hat{\mathcal{B}}_2)$ denote an optimal point to this minimization problem and define the corresponding reformulation substitute for B as $\acute{B} := B + \mathrm{diag}^*(\hat{d}_{\scriptscriptstyle B}) + v_{\scriptscriptstyle B}e^T + ev_{\scriptscriptstyle B}^T$. If $v_{\scriptscriptstyle B}$ is computed as in (4.9), then $\acute{B} = \hat{\mathcal{B}}_1 - \hat{\mathcal{B}}_2$. By Lemma 4.6, it is also apparent that the solution to problem (4.13) is unique and that $\hat{\mathcal{B}}_1$ and $\hat{\mathcal{B}}_2$ can be obtained by using the spectral decomposition of $\acute{B}$, as defined in (3.9). Obviously, the derived reformulation is designed to tighten the feasible set of the respective QAP_{MS} instance.

Let us consider some more arguments in favor of the presented reformulations. Numerical tests have shown that in most cases the two discussed approaches lead to improved bounds when compared to the results for the original formulation. Their major difference lies in the design for the respective relaxation techniques. The matrix splitting based relaxation benefits more from the reformulation instances derived via (4.9) and (4.13). The matrix lifting based relaxation, on the other hand, performs better when applied to the QAP reformulations computed via (4.9) and (4.12). Another major difference can be found in the accompanying computational aspects. Due to their explicit formulas, the parameters for the first described QAP reformulation can be computed significantly faster and more accurate. On the other hand, when it comes to the computation of the actual relaxation instance the second reformulation approach may have a significant advantage over the first one.

Denote by $\acute{\lambda}_1, \acute{\lambda}_2, \ldots, \acute{\lambda}_n$ the eigenvalues of the respective substitute $\acute{B}$. The first reformulation is designed to minimize the corresponding sum of squares $\sum_{i=1}^n \acute{\lambda}_i^2$. The utilization of problem (4.13), on the other hand, leads to the minimization of a weighted sum of their absolute values $\sum_{i=1}^k w_2|\acute{\lambda}_i| + \sum_{i=k+1}^n w_1|\acute{\lambda}_i|$, where k is the index which separates negative from nonnegative eigenvalues, i.e. $\acute{\lambda}_1 \leqslant \ldots \leqslant \acute{\lambda}_k < 0 \leqslant \acute{\lambda}_{k+1} \leqslant \ldots \leqslant \acute{\lambda}_n$. The latter minimization objective tends to produce sparser solutions, which turns out to be a very beneficial property.

We can make use of the decreased rank of $\acute{B}$ by reducing the respective SDP relaxations. The corresponding reduction scheme for the matrix-splitting based relaxation framework has already been emphasized in [84]. Let the matrices B_+ and B_- have ranks n_+ and n_-, respectively. There exist matrices $L_{B_+} \in \mathcal{M}^{n,n_+}$ and $L_{B_-} \in \mathcal{M}^{n,n_-}$ such that $B_+ = L_{B_+} L_{B_+}^T$ and $B_- = L_{B_-} L_{B_-}^T$. Then, without affecting the actual feasible set, (3.10b) can be replaced by

$$\begin{bmatrix} I_{(n_\diamond)} & L_{B_\diamond}^T X^T \\ X L_{B_\diamond} & Y_\diamond \end{bmatrix} \in \mathcal{S}_+^{n+n_\diamond} \qquad \text{for} \quad \diamond \in \{+,-\}. \tag{4.14}$$

In that context, we prefer reformulations that produce substitutes $\acute{B}_\diamond$ with lower ranks. For the usage in combination with *MS*, we therefore strongly recommend the application of reformulation instances obtained via (4.9) and (4.13).

At this point, it is worth mentioning that there is an efficient way to approximate the second QAP reformulation. One can avoid the computational overhead of problem (4.13) by requiring that all components of d_B are identical, i.e. there exists some $\beta \in \mathbb{R}$ such that $d_B = \beta e$:

$$\begin{aligned} \inf_{\beta \in \mathbb{R},\ \mathbf{B}_1, \mathbf{B}_2 \in \mathcal{S}_+^n} \quad & w_1 \operatorname{tr}(\mathbf{B}_1) + w_2 \operatorname{tr}(\mathbf{B}_2) \\ \text{s. t.} \quad & (I - \tfrac{1}{n} E)\,(B + \beta I)\,(I - \tfrac{1}{n} E) = \mathbf{B}_1 - \mathbf{B}_2. \end{aligned} \tag{4.15}$$

Apparently, the matrices $(I - \frac{1}{n}E)\,(B + \beta I)\,(I - \frac{1}{n}E)$ are simultaneously diagonalizable for each $\beta \in \mathbb{R}$. To obtain a solution to problem (4.15), it is therefore sufficient to compute the eigenvalue decomposition of $(I - \frac{1}{n}E)B(I - \frac{1}{n}E)$ and find the β that minimizes the weighted sum over the absolute values of the corresponding eigenvalues. Since β acts as a shift to all eigenvalues except the one that corresponds to the eigenvector $\frac{1}{\sqrt{n}}e$, the computational effort reduces to a weighted 1-norm minimization over these eigenvalues. An efficient procedure to solve this problem can be realized with an effort not greater than the identification of the weighted median of $n-1$ real values.

Requiring d_B to be a multiple of e has the big advantage of a significantly reduced computational overhead and a higher accuracy of the calculated parameters. On top of this, numerical examples from the QAP library [18] suggest that the solutions to problem (4.15) provide good approximations for the parameters computed via problem (4.13). In many cases the results are indeed identical. If not utilized for the actual relaxation

instance, the so derived approximation serves at least as a good initial point to the original minimization problem.

For the application of problem (4.13), it is necessary to determine appropriate weighting coefficients w_1 and w_2. Why do we not just minimize the sum of traces of the splitting parts B_1 and B_2? To answer this question, consider the problem

$$\begin{aligned} \inf_{X\in\mathcal{D}^n,\, Y_1, Y_2\in\mathcal{S}^n_+} \quad & \langle \acute{A}, Y_1 - Y_2\rangle + \langle \acute{C}, X\rangle \\ \text{s. t.} \quad & Y_i e = X\acute{B}_i e, \quad \operatorname{tr}(Y_i) = \operatorname{tr}(\acute{B}_i), \qquad i = 1, 2. \end{aligned} \tag{4.16}$$

If we assume that e lies in the kernel of the matrices $\acute{A}, \acute{B}_1, \acute{B}_2$ of the reformulated QAP instance $(\acute{A}, \acute{B}_1 - \acute{B}_2, \acute{C})$, then the optimal objective to this problem is

$$\lambda_{\min}(\acute{A}) \operatorname{tr}(\acute{B}_1) - \lambda_{\max}(\acute{A}) \operatorname{tr}(\acute{B}_2) + \min_{X\in\Pi^n} \langle \acute{C}, X\rangle.$$

To obtain strong lower bounds, it is therefore beneficial to utilize positive semidefinite splittings with traces $\operatorname{tr}(\acute{B}_1)$ and $\operatorname{tr}(\acute{B}_2)$ which are counterbalanced to the extreme eigenvalues of $\acute{A}$.

Due to tighter constraints in the relaxation QAP_{MS}, the actual situation is a good deal better. It is nevertheless advantageous to consider the distribution of the eigenvalues of $\acute{A}$. Let the reformulated parameter matrix $\acute{A}$ satisfy $\acute{A}e = \operatorname{diag}(\acute{A}) = 0$ and denote its eigenvectors by $\acute{\mu}_1, \ldots, \acute{\mu}_n$. For the computation of the weighting coefficients, we utilize the following formulas

$$w_1 = \left(\sum_{i:\, \acute{\mu}_i < 0} |\acute{\mu}_i|^p \right)^{\frac{1}{p}} \qquad \text{and} \qquad w_2 = \left(\sum_{i:\, \acute{\mu}_i > 0} |\acute{\mu}_i|^p \right)^{\frac{1}{p}}, \tag{4.17}$$

where p is used as a threshold between the equally weighted norm minimization ($p = 1$) and the - in respect of the extreme eigenvalues of $\acute{A}$ - completely counterbalanced norm minimization ($p = \infty$). In the actual implementation, we use an intermediate weighting and set $p = 4$.

The last arguments have all been in favor of the second reformulation approach. Nevertheless, also the first approach has its raison d'être. Besides the better results in combination with the relaxation ML, it also shows a beneficial interaction with some of the constraints presented in Section 6.1. In the attempt to ensure a fair comparison between the discussed relaxation frameworks, we want to avoid the utilization of multiple

reformulation approaches. For this purpose, we design a compromise between the two presented QAP reformulations. In the actual implementation, we obtain the parameter vector d_B as a part of the solution vector to the following problem:

$$\begin{aligned} \inf_{d_B \in \mathbb{R}^n,\ \mathcal{B}_1, \mathcal{B}_2 \in \mathcal{S}^n_+} \quad & \left\|\!\left| w_1 \mathcal{B}_1 + w_2 \mathcal{B}_2 \right|\!\right\|_{\mathrm{T}} + \left\|\!\left| w_1 \mathcal{B}_1 + w_2 \mathcal{B}_2 \right|\!\right\|_{\mathrm{F}} \\ \text{s. t.} \quad & (I - \tfrac{1}{n} E) \left(B + \operatorname{diag}^*(d_B) \right) (I - \tfrac{1}{n} E) = \mathcal{B}_1 - \mathcal{B}_2. \end{aligned} \tag{4.18}$$

The respective parameter vector v_B is still compliant with (4.9). The obtained QAP reformulation is the result of a trade-off between a minimal trace-norm of $\acute{B}$, a counterbalanced eigenvalue distribution in consideration of $\acute{A}$, and a minimization of the trace of $\acute{B}^2$. This is also reflected in the addressed properties, for instance, the tendency to keep a low-rank parameter matrix $\acute{B}$.

In consideration of the numerous references to the reformulated problem instances, from now on, we simply assume that (A, B, C) states the QAP instance that is obtained from the original problem by applying the considered QAP reformulation. The rarely used original formulation is instead referred to as $(\mathring{A}, \mathring{B}, \mathring{C})$. This adaption leads to a simplified notation. Unless otherwise stated, the parameters for the applied QAP reformulation scheme are

$$d_A = \frac{2}{n-2} \mathring{A} e - \frac{\langle \mathring{A}, E \rangle}{(n-1)(n-2)} e, \tag{4.19a}$$

$$v_A = w_A = -\frac{1}{2} d_A, \tag{4.19b}$$

$$d_B = \hat{d}_B \quad \text{optained via problem (4.18) with weighting from (4.17)}, \tag{4.19c}$$

$$v_B = w_B = \frac{\langle \mathring{B}, E \rangle + \langle d_B, e \rangle}{2n^2} e - \frac{1}{n} \left(\mathring{B} e + d_B \right). \tag{4.19d}$$

Although the changes on the data matrix $\mathring{A}$ do not affect the considered relaxation frameworks, due to certain advantages over the original formulation, we still apply the *EVB1* based reformulation to this matrix. One advantage is that additional reformulations of B do not change the coordinates in C, hence for arbitrary vectors $d, v, w \in \mathbb{R}^n$ the instances $(A, B + \operatorname{diag}^*(d) + ve^T + ew^T, C)$ describe QAP reformulations for the same problem. Another benefit is its suitability for the computation of weighting parameters via (4.17).

Summing up and reiterating this important point: from now on, the QAP instance (A, B, C) states a certain reformulation to the respective original problem $(\mathring{A}, \mathring{B}, \mathring{C})$. In

respect of Definition 4.2, these problems are related via

$$A = \mathring{A} + \overset{*}{\operatorname{diag}}(d_{A}) + v_{A}e^{T} + ev_{A}^{T}, \tag{4.20a}$$

$$B = \mathring{B} + \overset{*}{\operatorname{diag}}(d_{B}) + v_{B}e^{T} + ev_{B}^{T}, \tag{4.20b}$$

$$C = \mathring{C} - d_{A}d_{B}^{T} - 2\acute{A}ev_{B}^{T} - 2v_{A}e^{T}B, \tag{4.20c}$$

with $d_{A}, v_{A}, d_{B}, v_{B}$ being taken from (4.19).

The problem instances (A, B, C) and $(\mathring{A}, \mathring{B}, \mathring{C})$ have the same solution set, which is easily verified by

$$\forall X \in \Pi^{n}: \quad \operatorname{tr}(AXBX^{T} + CX^{T}) = \operatorname{tr}(\mathring{A}X\mathring{B}X^{T} + \mathring{C}X^{T}).$$

In addition to their symmetry, the matrices A and B always satisfy $\operatorname{diag}(A) \equiv 0$, $Ae \equiv 0$, and $Be \equiv 0$. Using the reformulated problem instance (A, B, C) as input for the low-dimensional relaxation frameworks usually leads to better results.

4.1.2. Non-redundant positive semidefinite matrix splitting

In Subsection 4.1.1, we presented well argued reformulation approaches which are designed for tightening the considered SDP relaxations. Though the focus was particularly on the matrix splitting based relaxation framework *MS*, the SDP constraint of *ML* was considered as well. We derived the QAP reformulation in (4.20) which gives a good compromise for the different relaxation techniques. In that context, we also discussed strong criteria for the choice of the positive semidefinite matrix splitting scheme defined in (3.9). Peng et al. [84] followed a different approach to address the issue of finding appropriate splitting schemes. They introduced the notion of redundant and non-redundant matrix splitting. Based on their observations, they designed two new splitting schemes. The stronger of the corresponding frameworks outperforms the relaxations introduced in [83] for most instances from the QAP library [18].

The author believes that this is a good moment for a more detailed explanation of the chosen positive semidefinite matrix splitting scheme. This subsection shall serve this purpose by giving a brief review of the matrix splitting schemes introduced in [83, 84] and presenting an alternative interpretation of these. For the latter, we reveal connections between the applied matrix splitting schemes and QAP reformulations.

Let us start with a short interlude in 'non-redundant positive semidefinite matrix splitting'.

Definition 4.7. *A positive semidefinite splitting (B_1, B_2) to a matrix B is said to be redundant if there exists a nonzero positive semidefinite matrix R, such that*

$$B = B_1 - B_2, \quad B_1 - R \in \mathcal{S}_+, \quad B_2 - R \in \mathcal{S}_+. \tag{4.21}$$

If $R \equiv 0$ is the only feasible matrix that is positive semidefinite and satisfies (4.21)*, the splitting is said to be non-redundant.*

In respect of a matrix splitting based SDP relaxation such as MS_0, Peng et al. demonstrated the general advantage of non-redundant matrix splittings over redundant ones, see [84, Theorem 1]. Roughly speaking, the theorem states that for any redundant positive semidefinite matrix splitting there exists a non-redundant splitting which leads to a tighter relaxation. Even though additional constraints on the respective variables may change this circumstance, the absence of redundancies in a positive semidefinite matrix splitting is a good indicator for a beneficial splitting scheme.

In [84, Theorem 2], it was proved that the matrix pair defined in (3.9) states a non-redundant positive semidefinite matrix splitting. It is the same scheme as the one used in [83] by Peng, Mittelmann, and Li. Lemma 4.6 gives an even stronger argument for the application of the matrix splitting approach based on the spectral decomposition. The positive semidefinite pair given in (3.9) states a minimizer for the objective $w_1 |||\boldsymbol{B}_1||| + w_2 |||\boldsymbol{B}_2|||$ regardless of the choice of the unitarily invariant matrix $|||\cdot|||$ and the weighting coefficients w_1, w_2. In this context, it is striking to see how the framework *SDRMS-SUM* from [84] outperforms the relaxation *SDRMS-SVD* discussed in the same paper but originally introduced in [83]. Even the framework *SDRMS-ONE* (also given in [84]) is not completely dominated by *SDRMS-SVD*, though it is based on a positive semidefinite matrix splitting which is usually redundant. How are these results reconcilable with our arguments for the splitting scheme used in MS_0?

The answer to this question has surprisingly little to do with a redundant or non-redundant positive semidefinite matrix splitting scheme. Consider, for instance, the minimal trace sum-matrix splitting approach that is used as the basis for *SDRMS-SUM*.

The corresponding splitting is obtained as the solution to

$$\begin{aligned} \inf_{d,v\in\mathbb{R}^n,\, B_2\in\mathcal{S}_+^n} \quad & \operatorname{tr}(B_2) \\ \text{s. t.} \quad & B + \operatorname{diag}^*(d) + ve^T + ev^T = -B_2. \end{aligned} \tag{4.22}$$

Interestingly, a closer look on the obtained solution (d, v, B_2) reveals that the corresponding matrix pair $(-\operatorname{diag}^*(d) - ve^T - ev^T, B_2)$ does not necessarily state a positive semidefinite matrix splitting to B. There are many QAP instances for which $\operatorname{diag}^*(d) + ve^T + ev^T$ is indefinite. The concept of non-redundant matrix splitting is therefore not applicable, at least not in a straightforward manner. This is where QAP reformulations come into consideration. With the parameters d and v, it is possible to construct three different symmetric QAP reformulations that involve individual interpretations of the splitting scheme:

$$\acute{B} = B + \operatorname{diag}^*(d) \qquad \Longrightarrow \acute{B} = \big(-ve^T - ev^T\big) - B_2, \tag{4.23a}$$

$$\acute{B} = B + ve^T + ev^T \qquad \Longrightarrow \acute{B} = \big(-\operatorname{diag}^*(d)\big) - B_2, \tag{4.23b}$$

$$\acute{B} = B + \operatorname{diag}^*(d) + ve^T + ev^T \qquad \Longrightarrow \acute{B} = 0_{(n,n)} - B_2. \tag{4.23c}$$

In general, neither $(-\operatorname{diag}^*(d))$ nor $(-ve^T - ev^T)$ have to be positive semidefinite. For many instances of the QAP library [18], these matrices are indefinite. The only interpretation for which the respective splitting always complies with Definition 3.1 is the last one.

In regard to the interpretation given in (4.23c), the minimal trace one-matrix splitting [84, Section 3.1] is non-redundant as well. The respective splitting scheme simply corresponds to a different QAP reformulation. For $\acute{B} = 0_{(n,n)} - B_2$, it is moreover apparent that $(0_{(n,n)}, B_2)$ states the unique non-redundant positive semidefinite splitting to $\acute{B}$. The compliance with the matrix pair in (3.9) is evident.

These observations lead to a new interpretation of the frameworks *SDRMS-ONE* and *SDRMS-SUM*. The programs are specific versions of QAP_{MS} and can be derived from *SDRMS-SVD* by applying the respective QAP reformulations. *SDRMS-SUM* is superior to *SDRMS-ONE* because it utilizes a more beneficial reformulation.

The results given in Subsection 4.1.1 suggest an even greater benefit from the reformulation defined in (4.20). Though this does not account to every QAP instance, numerical results reinforce this suggestion. Nevertheless, it should also be mentioned that the optimal combination of a QAP reformulation and a positive semidefinite matrix splitting scheme depends on many more factors. The consideration of the eigenvalue

distribution of the reformulated parameter matrix A is a good start. However, the update of A is more an initial guess than an optimized reformulation that considers the mutual correlation between the updates for both matrices A and B. How the reformulated coefficient matrix C affects relaxation results has not been taken into account at all. Hence, the presented reformulation approach still leaves room for improvement.

4.1.3. Commutation of data matrices

The concept of reformulating quadratic assignment problems is not limited to what we called QAP reformulations in the last subsections. Another possibility to reformulate a problem instance (A, B, C) would be a counterbalanced scaling of the parameter matrices A and B, i.e. $(\grave{A}, \grave{B}, \grave{C}) = (\alpha A, \alpha^{-1} B, C)$ for some $\alpha \in \mathbb{R}_{\neq 0}$. This kind of reformulation is not considered in this section because the objective values of the corresponding SDP relaxations are not affected by scaling. In a different context, we will come back to this reformulation, demonstrating its value for resolving numerical difficulties that occur in the solution procedure.

This subsection is about a very basic reformulation approach, which - as elementary as it is - often has a significant influence on the obtained bounds. We mean the commutation of the coefficient matrices A and B. It is very easy to see that the instance $(\grave{A}, \grave{B}, \grave{C}) = (B, A, C^T)$ describes an equivalent problem to the actual QAP. The only necessary adaptation is that any optimal assignment $\hat{X}$ is transposed to an optimal permutation matrix of the reformulated problem instance, and vice versa.

Once again, the vector lifting approach demonstrates its strength. Neither the QAP reformulations defined in (4.2) nor the commutation of A and B affect the lower bounds computed via VL_0. On the other hand, the permutation of the data matrices often has a significant influence on the results of the other SDP relaxations.

The most obvious and safest method to decide whether A and B shall be exchanged or not is the numerical test of both formulations. In most cases, however, solving the SDP relaxation two times for each problem instance is not necessary. Different measurements for the natural advantage of a certain data matrix commutation can be evaluated with significantly smaller computational costs. A useful indicator for the tightness of our SDP relaxations is the relative distance to the norm-wise closest sum-matrix:

$$\delta_{SUM}(B) := \min \left\{ \frac{\| \operatorname{off}(B + ve^T + ev^T) \|}{\| \operatorname{off}(B) \|} \,\middle|\, v \in \mathbb{R}^n \right\}. \tag{4.24}$$

An explicit formula for the vector $\hat{v}$ that minimizes this norm fraction can be derived by applying (4.9) and (4.12):

$$\hat{v} = \frac{1}{n-2}\left(Be - \frac{\langle B, E\rangle}{2(n-1)}e\right).$$

If $\delta_{SUM}(A) < \delta_{SUM}(B)$, it is typically beneficial to interchange the matrices A and B, otherwise it seems to be favorable to keep the original problem formulation. This statement is reasoned by the observation that the diagonal elements as well as the sum-matrix part of the respective parameter matrix can be linearized. The difficult part is the quadratic remainder.

A significant drawback of the relative distance defined by (4.24) is the dependency on the formulation. It is possible to reformulate the QAP beforehand using the formula for $\hat{v}$ and its adaptation for the reformulation of A, such that $\delta_{SUM}(A) = \delta_{SUM}(B) = 1$. A suitable assumption for the applicability of this indicator is that the minimal off-diagonal elements of the corresponding matrices are equal to zero.

Another possibility to overcome this issue is to use a second indicator that exploits different norms on a fixed QAP reformulation:

$$\delta_{21}(B) := \frac{\|\operatorname{off}(B + \hat{v}e^T + e\hat{v}^T)\|_2}{\|\operatorname{off}(B + \hat{v}e^T + e\hat{v}^T)\|_1}. \tag{4.25}$$

Whenever $\delta_{21}(A)$ is considerably smaller than $\delta_{21}(B)$, we swap the data matrices. Only for the instances where these values are close to each other, i.e. $\delta_{21}(A) \approx \delta_{21}(B)$, we still test both formulations and choose the one that delivers stronger bounds. This commutation rule has already been used for the numerical results in Table 3.1.

4.2. Reduced relaxation frameworks

A closer examination of the level-0 relaxations considered in Section 3.1 reveals some redundancies. We use this subsection to point out how the number of equality constraints can be reduced by replacing them with a smaller number of equivalent conditions.

In [24], Klerk and Sotirov showed that the equality

$$\operatorname{tr}(\mathcal{Y}) - 2\langle e, x\rangle = -n, \tag{4.26}$$

together with the positive semidefiniteness condition in (3.3b) and the Gangster equations (3.3g), (3.3f), imply the validity of (3.3c), (3.3d), and (3.3e). The single condition (4.26) replaces $n^3 + n^2 + 2n$ equality constraints of the corresponding formulation QAP_{VL_0}.

For the reduction of the other SDP frameworks, we consider the following result.

Lemma 4.8. *The replacement of* (3.5d) *and* (3.10d) *with*

$$\langle E, Z \rangle = \langle E, B^2 \rangle, \tag{4.27}$$

and

$$\langle E\,,\, Y_+ + Y_- \rangle = \langle E\,,\, B_+ + B_- \rangle, \tag{4.28}$$

respectively, does not affect the feasible set of the corresponding relaxation frameworks. Furthermore, in accordance to the feasible set described by ES_0, the equality constraints in (3.11d) *are redundant.*

Proof. For the discussion of the first replacement, let (X, Y, Z) denote an arbitrary feasible point to the respective instance of QAP_{ML_0}. The conditions $X \in \mathcal{D}$ and (3.5d) imply

$$\langle E, Z \rangle = e^T Z e = e^T X B^2 e = e^T B^2 e = \langle E, B^2 \rangle.$$

Any feasible point to problem (3.5) therefore also satisfies (4.27).

In order to prove the other direction, let (X, Y, Z) denote an arbitrary feasible point to the respective instance of *ML* with the replaced equality constraint (4.27) instead of (3.5d). Define by

$$r_{_Y} := Ye - XBe \qquad \text{and} \qquad r_{_Z} := Ze - XB^2e$$

the residuals to the replaced equations in (3.5d). The SDP constraint in (3.5b) requires

$$\vartheta_{_Y}(\alpha) := \begin{pmatrix} \alpha e \\ -\alpha e \\ r_{_Y} \end{pmatrix}^T \begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \begin{pmatrix} \alpha e \\ -\alpha e \\ r_{_Y} \end{pmatrix}$$

to be nonnegative for any choice of $\alpha \in \mathbb{R}$. By positive semidefiniteness of Z and the diagonal equalities in (3.5c), we additionally have

$$\langle r_{Y}, Z r_{Y} \rangle \leqslant \|\!| Z |\!\|_2 \, \|r_{Y}\|^2 \leqslant \operatorname{tr}(Z) \, \|r_{Y}\|^2 = \operatorname{tr}(B^2) \, \|r_{Y}\|^2 .$$

Since

$$\begin{aligned} 0 \leqslant \vartheta_{Y}(\alpha) &= 2\alpha^2 n - 2\alpha^2 e^T X e + \big(2\alpha X B e - 2\alpha Y e + Z r_{Y}\big)^T r_{Y} \\ &= \big(-2\alpha r_{Y} + Z r_{Y}\big)^T r_{Y} \\ &\leqslant \big(-2\alpha + \operatorname{tr}(B^2)\big) \, \|r_{Y}\|^2 \end{aligned}$$

holds true for arbitrarily large values of α, the norm of r_{Y} needs to be equal to zero. This, in turn, implies the compliance with the original constraint $Ye = XBe$.

In the same way, we derive

$$\begin{aligned} 0 \leqslant & \begin{pmatrix} \alpha B e \\ \vec{0} \\ r_{Z} - \alpha e \end{pmatrix}^T \begin{bmatrix} I & X^T & B X^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \begin{pmatrix} \alpha B e \\ \vec{0} \\ r_{Z} - \alpha e \end{pmatrix} \\ &\leqslant -\alpha^2 e^T B^2 e + \alpha^2 e^T Z e + \big(-2\alpha + \operatorname{tr}(B^2)\big) \, \|r_{Z}\|^2 \\ &= \big(-2\alpha + \operatorname{tr}(B^2)\big) \, \|r_{Z}\|^2 , \end{aligned}$$

where the validity of the latter equality is due to (4.27). The immediate consequence of this observation is the compliance with the vector constraint $Ze = XB^2e$.

A very similar approach can be used to account the substitution (4.28) for (3.10d). The general idea is to exploit the non-negativity of

$$\sum_{\diamond \in \{+,-\}} \begin{pmatrix} \alpha e \\ r_{Y_\diamond} - \alpha e \end{pmatrix}^T \begin{bmatrix} B_\diamond & B_\diamond X^T \\ X B_\diamond & Y_\diamond \end{bmatrix} \begin{pmatrix} \alpha e \\ r_{Y_\diamond} - \alpha e \end{pmatrix}.$$

The remaining assertions follow the arguments used for the first replacement.

By the same operandi, it is possible to prove the redundancy of (3.11d). To this end, define

$$r_{Q_i} := Q_i e - X q_i q_i^T e \qquad \text{for} \quad i = 1, \ldots, n,$$

as well as

$$\vartheta_{Q}(\alpha) := \sum_{i=1}^{n} \begin{pmatrix} \alpha(q_i^T e) \\ r_{Q_i} - \alpha e \end{pmatrix}^T \begin{bmatrix} 1 & q_i^T X^T \\ X q_i & Q_i \end{bmatrix} \begin{pmatrix} \alpha(q_i^T e) \\ r_{Q_i} - \alpha e \end{pmatrix}.$$

Together with the equality constraints (3.11c), (3.11e) and the requirement $X \in \mathcal{D}$, the semidefiniteness condition in (3.11b) implies

$$\begin{aligned} 0 &\leqslant \vartheta_{Q}(\alpha) \\ &\leqslant \sum_{i=1}^{n} \Big(-\alpha^2 e^T q_i q_i^T e + \alpha^2 e^T Q_i e + \big(-2\alpha + \operatorname{tr}(Q_i) \big) \left\| r_{Q_i} \right\|^2 \Big) \\ &= (-2\alpha + 1) \sum_{i=1}^{n} \|r_{Q_i}\|^2, \end{aligned}$$

which validates the compliance with (3.11d). □

4.3. Level-1 relaxations

We conclude this chapter about reformulation techniques with a small summary of the discussed modifications. For this purpose, we present the level-1 versions of the corresponding frameworks and evaluate their applicability on the basis of a few numerical examples. The notation follows the one used in Subsection 4.1.1; in particular, (A, B, C) refers to the reformulated problem instances given in (4.20). This notation is also transfered to the spectral value decomposition of the respective parameter matrices, i.e. $A = \sum_{i=1}^{n} \mu_i p_i p_i^T$ and $B = \sum_{i=1}^{n} \lambda_i q_i q_i^T$.

We have already shown that the considered adaptations do not effect the feasible set of relaxation programs which are based on vector lifting. The level-1 version of QAP_{VL} is therefore equivalent to the level-0 version given in (3.3). Nevertheless, there are some significant changes to this relaxation, making it worth to present the framework VL_1:

$$\inf_{X\in\mathcal{M}^n,\, Y\in\mathcal{S}^n,\, \Upsilon\in\mathcal{S}^{n^2}} \quad \langle A, Y\rangle + \langle C, X\rangle \tag{4.29a}$$

$$\text{s. t.} \quad \begin{bmatrix} 1 & \operatorname{vec}(X)^T \\ \operatorname{vec}(X) & \Upsilon \end{bmatrix} \in \mathcal{S}_+, \tag{4.29b}$$

$$\operatorname{tr}(\Upsilon) = \langle E, X\rangle = n, \tag{4.29c}$$

$$\big(I\otimes(E-I) + (E-I)\otimes I\big)\circ\Upsilon = 0, \tag{4.29d}$$

$$(e\otimes I)^T\big(\Upsilon\circ(B\otimes E)\big)(e\otimes I) = Y. \tag{4.29e}$$

For improved convergence in the solving procedure, here we are applying the two equalities in (4.29c) instead of the single equality condition (4.26). Constraint (4.29e) is deduced from the identity

$$XBX^T = (e\otimes I)^T\left[(\operatorname{vec}(X)\operatorname{vec}(X)^T)\circ(B\otimes E)\right](e\otimes I). \tag{4.30}$$

Except for QAP_{ML}, the optimization variable Y is not required for the actual implementation of the individual programming problem. For reasons of clarity, we nevertheless make use of the same variables X and Y in all four relaxation programs. A positive side effect of this procedure is that the objective functions as well as several constraints which will be discussed in Section 6.1 have the same form in all considered relaxations. It is therefore sufficient to describe the related adaptations only once.

The level-1 versions of the other SDP relaxations are listed below. Firstly, the framework ML_1:

$$\inf_{X\in\mathcal{D}^n,\ Y,Z\in\mathcal{S}^n} \quad \langle A, Y\rangle + \langle C, X\rangle \tag{4.31a}$$

$$\text{s. t.} \quad \begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \in \mathcal{S}_+, \tag{4.31b}$$

$$\operatorname{diag}(Y) = X\operatorname{diag}(B), \quad \operatorname{diag}(Z) = X\operatorname{diag}(B^2), \tag{4.31c}$$

$$\langle E, Z\rangle = 0. \tag{4.31d}$$

Secondly, the relaxation based on matrix splitting:

$$\inf_{X\in\mathcal{D}^n,\ Y,Y_+,Y_-\in\mathcal{S}^n} \quad \langle A, Y\rangle + \langle C, X\rangle \tag{4.32a}$$

$$\text{s. t.} \quad \begin{bmatrix} I_{(n_\diamond)} & L_\diamond^T X^T \\ X L_\diamond & Y_\diamond \end{bmatrix} \in \mathcal{S}_+, \qquad \diamond \in \{+,-\}, \tag{4.32b}$$

$$\operatorname{diag}(Y_\diamond) = X \operatorname{diag}(B_\diamond), \qquad \diamond \in \{+,-\}, \tag{4.32c}$$

$$\langle E, Y_+\rangle = 0, \quad \langle E, Y_-\rangle = 0, \tag{4.32d}$$

$$Y_+ - Y_- = Y, \tag{4.32e}$$

where the matrices $L_\diamond \in \mathcal{M}^{n,n_\diamond}$ are obtained via some compact decomposition $B_\diamond = L_\diamond L_\diamond^T$. And finally, the "eigenspace" SDP relaxation:

$$\inf_{X\in\mathcal{D}^n,\ Q_1,\dots,Q_n,Y\in\mathcal{S}^n} \quad \langle A, Y\rangle + \langle C, X\rangle \tag{4.33a}$$

$$\text{s. t.} \quad \begin{bmatrix} 1 & q_i^T X^T \\ X q_i & Q_i \end{bmatrix} \in \mathcal{S}_+, \qquad i \in \{1,\dots,n\}, \tag{4.33b}$$

$$\operatorname{diag}(Q_i) = X \operatorname{diag}(q_i q_i^T), \qquad i \in \{1,\dots,n\}, \tag{4.33c}$$

$$\sum_{i=1}^n Q_i = I, \quad \sum_{i=1}^n \lambda_i Q_i = Y. \tag{4.33d}$$

The constraints in (4.31d) and (4.32d) emerge from the reformulation property $\langle E, B\rangle = \langle E, B_+\rangle = \langle E, B_-\rangle = 0$ and replace the corresponding vector equalities. If other QAP reformulations with different properties shall be used, the right-hand sides of these equality constraints have to be adapted accordingly. Apart from the introduction of the variable Y in all considered relaxation frameworks and the realization of the reduction approach from (4.14), the only noteworthy difference to their level-0 counterparts is the utilization of the discussed reformulation approach. The fact that this is the sole modification with an actual influence on the bounding quality is reflected in the selected numerical examples given in Table 4.1.

The bounds computed via VL_1 are, as expected, the same as the ones obtained via the corresponding level-0 version. On the other hand, the QAP reformulation from (4.20) helps to improve the bounds obtained by the other relaxations. The largest improvement is attained on the bounds that are computed via QAP_{MS}. In particular, the

Table 4.1.: Selected bounds for comparison of level-1 relaxations [R_{gap} in (%)]

Problem	VL_1	ML_1	MS_1	ES_1
Chr12a	**150.68**	230.74	312.34	195.71
Esc16b	**5.48**	10.15	5.65	**5.48**
Had14	**0.56**	7.35	3.53	1.56
LiPa20a	**0.72**	1.70	3.62	1.35
Nug12	**8.42**	18.21	17.80	10.11
Scr20	**24.26**	70.59	39.93	28.46
Tai17a	**10.27**	15.16	24.11	12.92

instances *Esc16b* and *Scr20* demonstrate the high benefit for this relaxation framework. In anticipation of the expected further improvements, these results are already quite promising. Nevertheless, for many QAP instances, the obtained bounds are still far away from being of practical use. Moreover, there are a few instances for which the bounds obtained via the respective level-1 relaxations are not improved in comparison to their level-0 counterparts.

Chapter 5.

New Relaxation Frameworks

5.1. New relaxation approaches

In the previous sections, we revisited four different SDP relaxations for the quadratic assignment problem. Though Theorem 3.2 exhibits correlations between these frameworks, the respective programs originate from considerable different relaxations strategies. This is reflected in the strongly differing computational expenses as well as the obtained lower bounds.

Due to their great number of variables, relaxations based on vector lifting techniques are usually too expensive for practical usage. Though the complexity of ES is a good deal smaller than the one of VL - about a factor n in the number of variables as well as in the number of equality constraints - the same statement also applies to the "eigenspace" SDP relaxation. The latter optimizes over a still large number of $\mathcal{O}(n^3)$ variables. On the other hand, regarding their tightness, the gap between the high-dimensional and the low-dimensional frameworks is all but not negligible. For a better efficiency, we either reduce the computational expense of the high-dimensional frameworks or increase the bounding quality of the low-dimensional ones. In Subsection 5.1.1, we provide approximation approaches to increase the efficiency of the relaxations discussed in Section 3.1. Subsequently, we introduce new SDP frameworks which extend the already presented relaxation techniques.

5.1.1. Reduction via approximation

In numerical tests, we observed that the applied QAP reformulation scheme often induces strongly differing traces of the positive and negative semidefinite parts of the reformulated B. In cases where the norm of one of these matrices $B_\diamond \in \{B_+, B_-\}$ falls below some relative threshold, i.e. $\|\!|B_\diamond|\!\|_2 \leqslant \varepsilon \|\!|B|\!\|_2$, we suggest a reduction of the matrix splitting based framework by removing the associated variable $Y_\diamond$. Due to its insignificant spectral norm, $Y_\diamond$ plays only a minor role for the quality of the relaxation framework, whereas its elimination strongly reduces the accompanied computational costs.

The removal of the corresponding matrix variable can be realized by exploiting a different QAP reformulation which is based on the minimal trace sum-matrix splitting introduced in [84]. The connection between QAP reformulations and this specific splitting approach has already been discussed in Subsection 4.1.2.

For a threshold $\varepsilon \ll 1$, suppose that $\|\!|B_+|\!\|_2 \leqslant \varepsilon \|\!|B|\!\|_2$. In this case, instead of deriving the reformulation parameters by solving problem (4.13), one may utilize the solution $(\hat{d}_B, \hat{v}_B)$ to the following maximization problem

$$\begin{array}{ll} \sup\limits_{d_B, v_B \in \mathbb{R}^n} & \langle e, d_B + 2v_B \rangle \\ \text{s. t.} & -\mathring{B} - \overset{*}{\operatorname{diag}}(d_B) - v_B e^T - e v_B^T \in \mathcal{S}_+. \end{array} \tag{5.1}$$

Any non-redundant positive semidefinite matrix splitting of the reformulated data matrix $\acute{B} = \mathring{B} + \overset{*}{\operatorname{diag}}(\hat{d}_B) + \hat{v}_B e^T + e\hat{v}_B^T$ leads to $\acute{B}_+ = 0_{(n,n)}$ and $\acute{B}_- = \acute{B}$. The variable Y_+ as well as the related constraints thus become irrelevant. The elimination of these accompanies a drastic improvement of the efficiency and is, in the same manner, also applicable for the case $\|\!|B_-|\!\|_2 \leqslant \varepsilon \|\!|B|\!\|_2$.

Since the reduction approach from above is beneficial only in the presence of the described circumstances and only in combination with the frameworks based on positive semidefinite matrix splitting, we need to find other reduction strategies with more general applicability. An alternative approximation approach is given by Peng, Mittelmann, and Li [83]. In the reduced version of their matrix splitting based SDP relaxation, they replaced the semidefiniteness condition in (3.10b) by the plain conditions $Y_+, Y_- \in \mathcal{S}_+$. They observed that this substitution decreases the computing times significantly, whereas the the computed bounds are decreasing marginally. We follow this realization and construct similar constraint approximations for the discussed relaxation frameworks.

If we think of the matrices B_+ and B_- as linear operators, we may interpret the constraint reduction from (3.10b) to (4.14) in respect of a projection onto the images of these operators. For all matrices $V_\diamond \in \mathcal{M}^{n,n_\diamond}$ whose column vectors span a superset of the image of the respective operator, i.e. $\mathcal{R}(V_\diamond) \supseteq \mathcal{R}(B_\diamond)$, the semidefiniteness conditions

$$\begin{bmatrix} V_\diamond & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} B_\diamond & B_\diamond X^T \\ X B_\diamond & Y_\diamond \end{bmatrix} \begin{bmatrix} V_\diamond & 0 \\ 0 & I \end{bmatrix} \in \mathcal{S}_+ \tag{5.2}$$

are equivalent. Naturally, the inclusion $\mathcal{R}(V_\diamond) \supseteq \mathcal{R}(B_\diamond)$ requires the dimension $n_\diamond$ to be at least as large as the rank of the corresponding matrix $B_\diamond$. And of course, it is possible to find matrices $V_\diamond$ that have exactly $n_\diamond = \operatorname{rank}(B_\diamond)$ columns and still satisfy the condition on their column spaces. The constraints in (4.14), for instance, are realized using the transformation matrices $V_\diamond = (L_\diamond^\dagger)^T$ for $\diamond \in \{+,-\}$. The identity of the column spaces $\mathcal{R}(L_\diamond)$ and $\mathcal{R}(B_\diamond)$ validates the compliance with the demanded inclusion $\mathcal{R}(V_\diamond) \supseteq \mathcal{R}(B_\diamond)$.

In the attempt to reduce the dimensions even further, we are looking for low-rank approximations $\tilde{V}_\diamond \in \mathcal{M}^{n,\tilde{n}_\diamond}$, where $\tilde{n}_\diamond < \operatorname{rank}(B_\diamond)$ and $\mathcal{R}(V_\diamond) \subset \mathcal{R}(B_\diamond)$. Appropriate choices for $\tilde{V}_\diamond$ can be found by inspecting the corresponding Schur complement conditions:

$$Y_\diamond \succeq X B_\diamond \tilde{V}_\diamond (\tilde{V}_\diamond^T B_\diamond \tilde{V}_\diamond)^\dagger \tilde{V}_\diamond^T B_\diamond X^T \qquad \text{for} \quad \diamond \in \{+,-\}.$$

For a good trade-off between speed and quality, the matrices $\tilde{V}_\diamond$ shall have low dimensions $\tilde{n}_\diamond$ and - in respect of some unitarily invariant matrix norm - involve small residuals

$$\| B_\diamond - B_\diamond \tilde{V}_\diamond (\tilde{V}_\diamond^T B_\diamond \tilde{V}_\diamond)^\dagger \tilde{V}_\diamond^T B_\diamond \| \leqslant \varepsilon \| B \| \qquad \text{for} \quad \diamond \in \{+,-\}, \tag{5.3}$$

where ε denotes a bound for the relative approximation error.

In consideration of this demand, we make use of the spectral decomposition of the respective parameter matrices. More specifically, we utilize Theorem 2.4. Although it is possible to adjust the following procedure for arbitrary choices of unitarily invariant matrix norms, for reasons of simplicity, let us assume that the inequalities in (5.3) are interpreted with respect of the spectral norm. For the set of eigenvalues of B, define the index sets

$$\Omega_+^\varepsilon := \{ i \mid \lambda_i > \varepsilon \| B \|_2 \}, \qquad \Omega_-^\varepsilon := \{ i \mid \lambda_i < -\varepsilon \| B \|_2 \} \tag{5.4a}$$

as well as their union and its complement

$$\Omega^\varepsilon := \{i \mid |\lambda_i| > \varepsilon |\!|\!| B |\!|\!|_2\}, \qquad \mho^\varepsilon := \{i \mid |\lambda_i| \leqslant \varepsilon |\!|\!| B |\!|\!|_2\}. \tag{5.4b}$$

Furthermore, for some index set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$, additionally define

$$Q_\Omega := [q_{\omega_1}, \ldots, q_{\omega_k}] \qquad \text{and} \qquad \Lambda_\Omega := \operatorname{diag}^*([\lambda_{\omega_1}, \ldots, \lambda_{\omega_k}]). \tag{5.5}$$

In the actual implementation, a minimal rank reduction that complies with (5.3) is realized by applying $\tilde{V}_\diamond = Q_{\Omega^\varepsilon_\diamond} \Lambda^{-1}_{\Omega^\varepsilon_\diamond}$, from which we derive the SDP constraints

$$\begin{bmatrix} \Lambda^{-1}_{\Omega^\varepsilon_\diamond} & Q^T_{\Omega^\varepsilon_\diamond} X^T \\ X Q_{\Omega^\varepsilon_\diamond} & Y_\diamond \end{bmatrix} \in \mathcal{S}^{n+|\Omega^\varepsilon_\diamond|}_+ \qquad \text{for} \quad \diamond \in \{+,-\}. \tag{5.6}$$

For $\varepsilon = 0$, the conditions in (5.6) are equivalent to (4.14) as well as (3.10b), whereas these constraints reduce to $Y_+, Y_- \in \mathcal{S}_+$ for any $\varepsilon \geqslant 1$. In this respect, (5.6) replaces the original SDP conditions of MS_0 and provides an additional threshold parameter for weighting quality versus speed.

By a similar procedure, it is also possible to decrease the dimension of the SDP constraint of relaxation ML_0. We follow the example of (5.2) and reduce the constraint in (3.5b) by using some matrix $\tilde{V} \in \mathcal{M}^{n,\tilde{n}}$:

$$\operatorname{diag}^*(\tilde{V}, I, I)^T \begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \operatorname{diag}^*(\tilde{V}, I, I) \in \mathcal{S}^{2n+\tilde{n}}_+. \tag{5.7}$$

A suitable choice for $\tilde{V}$ can be constructed by concatenation of $\tilde{V}_+$ and $\tilde{V}_-$ from above, i.e. $\tilde{V} = \begin{bmatrix} \tilde{V}_+ & \tilde{V}_- \end{bmatrix}$.

Here it is worth mentioning that for $\tilde{n} < n$ the reduction via (5.7) usually results in an approximation of the original constraint. In contrast to the semidefinite substitutes in (5.2), this observation is independent from the relation between $\mathcal{R}(\tilde{V})$ and $\mathcal{R}(B)$. For the construction of an equivalent to (3.5b) one needs to apply the stricter requirement $\mathcal{R}(\tilde{V}) \supseteq \mathcal{R}(B) \cup \mathcal{R}(XB)$, which - except for the trivial case that all elements B are equal to zero - necessitates dimensions $\tilde{n} \geqslant n - 1$. Even the slight reduction from $\mathcal{S}^{3n}_+$ to $\mathcal{S}^{3n-1}_+$ is attainable only if e lies in the kernel of B. In different numerical tests, we further

observed that the approximation of condition (3.5b) does not work as efficient as the substitution (5.2) in framework *MS*. Nevertheless, depending on the circumstances, one may benefit from the use of (5.7) as a substitute for (4.31b).

After having seen two different examples for approximation strategies that may be used to construct more efficient relaxation frameworks, we pay our attention to the key aspect of this subsection: the reduction of the "eigenspace" SDP relaxation. A beneficial way to exploit a possibly reduced rank of the reformulated parameter matrix B is the unification of the nullspace. Assume that the index sets Ω^ε and $\mho^\varepsilon$ satisfy the definitions in (5.4b). For $\varepsilon = 0$, the constraints

$$\begin{bmatrix} 1 & q_i^T X^T \\ X q_i & Q_i \end{bmatrix} \in \mathcal{S}_+^{n+1} \qquad \text{for} \quad i \in \Omega^\varepsilon, \tag{5.8a}$$

$$\begin{bmatrix} I_{(|\mho^\varepsilon|)} & Q_{\mho^\varepsilon}^T X^T \\ X Q_{\mho^\varepsilon} & I - \sum\limits_{i \in \Omega^\varepsilon} Q_i \end{bmatrix} \in \mathcal{S}_+^{n+|\mho^\varepsilon|} \tag{5.8b}$$

describe a moderate relaxation of the original conditions in (3.11b).

For many instances from the QAP library [18], the unification of the nullspace reduces the number of variables significantly. On the other hand, the effect on the respective optimal values is negligible for all tested problems. This observation suggests a further reduction of the SDP constraints via increasing the approximation tolerance. The direct utilization of (5.8) for $\varepsilon > 0$, however, introduces an approximation error that requires special consideration. It is possible to estimate the introduced error by applying another bounding procedure to the remainder QAP. Alternatively, one may incorporate the respective error term by introducing additional programming variables. Either way, in the authors opinion, the effort is scarcely beneficial. Usually, the approximation error increases too fast against the descent of the number of semidefinite conditions.

The unification of SDP constraints for every cluster of eigenvalues seems to be a significantly better approach for the reduction of ES_1. For many instances of the parameter matrix B, it is possible to construct tight approximates $\tilde{B}$ that have very few different eigenvalues. This, in turn, allows a significant reduction of the considered relaxation framework. Let the reformulated data matrix B be split into an approximate $\tilde{B}$ and a residual R. For the relaxation of the remainder term $\operatorname{tr}(AXRX^T)$, we follow the matrix splitting approach and denote by R_+ and R_- the positive and negative semidefinite

part of R, respectively, such that

$$B = \tilde{B} + R_+ - R_-, \qquad R_+, R_- \in \mathcal{S}_+. \tag{5.9}$$

In the following, we assume that $\tilde{B}$ has k distinct eigenvalues. We denote the set of these values by $\{\tilde{\lambda}_{1\star}, \ldots, \tilde{\lambda}_{k\star}\}$ and - in respect of the corresponding multiset $\{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n\}$ - define the index sets

$$\Phi_i := \{j \mid \tilde{\lambda}_j = \tilde{\lambda}_{i\star}\} \qquad \text{for clusters} \quad i = 1, \ldots, k. \tag{5.10}$$

Finally, we combine the relaxation approaches of QAP_{ES} and QAP_{MS} to construct the "eigenspace cluster" SDP relaxation, referred to as QAP_{ESC} or simply *ESC*:

$$\inf_{X \in \mathcal{D}^n,\ F_+, F_-, U_1, \ldots, U_k, Y \in \mathcal{S}^n} \quad \langle A, Y \rangle \ + \ \langle C, X \rangle \tag{5.11a}$$

$$\text{s. t.} \quad \begin{bmatrix} I_{(|\Phi_i|)} & \tilde{Q}_{\Phi_i}^T X^T \\ X \tilde{Q}_{\Phi_i} & U_i \end{bmatrix} \in \mathcal{S}_+, \qquad 1 \leqslant i \leqslant k, \tag{5.11b}$$

$$\begin{bmatrix} I_{(n_\diamond)} & L_{R_\diamond}^T X^T \\ X L_{R_\diamond} & F_\diamond \end{bmatrix} \in \mathcal{S}_+, \qquad \diamond \in \{+, -\}, \tag{5.11c}$$

$$\begin{aligned} &\operatorname{diag}(U_i) = X \operatorname{diag}(\tilde{Q}_{\Phi_i} \tilde{Q}_{\Phi_i}^T), \qquad 1 \leqslant i \leqslant k, \\ &\operatorname{diag}(F_\diamond) = X \operatorname{diag}(R_\diamond), \qquad \diamond \in \{+, -\}, \end{aligned} \tag{5.11d}$$

$$\langle E, F_+ + F_- \rangle = \langle E, R_+ + R_- \rangle, \tag{5.11e}$$

$$\sum_{i=1}^k U_i = I, \quad \sum_{i=1}^k \tilde{\lambda}_{i\star} U_i + F_+ - F_- = Y, \tag{5.11f}$$

where $\tilde{Q} := [\tilde{q}_1, \ldots, \tilde{q}_n]$ denotes an orthogonal matrix consisting of a possible set of eigenvectors of $\tilde{B}$, and $L_{R_\diamond} \in \mathcal{M}^{n, n_\diamond}$ are derived via some compact decomposition of the residual matrices: $R_\diamond = L_{R_\diamond} L_{R_\diamond}^T$.

In order to obtain a suitable approximate $\tilde{B}$, we apply a k-median clustering algorithm [51] to the eigenvalues $\{\lambda_i\}$ of the parameter matrix B. The approximate $\tilde{B}$ is then constructed by replacing all eigenvalues of B with the computed centers of the corresponding clusters. The number of clusters is chosen just large enough to satisfy

$$|||B - \tilde{B}|||_2 \leqslant \varepsilon |||B|||_2. \tag{5.12}$$

Once more, ε is not only the bound for the relative approximation error but also serves as a threshold parameter for weighting quality versus speed. As a side effect of the applied approximation procedure, all four matrices $B, \tilde{B}, R_+, R_-$ are simultaneously diagonalizable. Other splitting schemes, that do not comply with this characteristic or require different reformulation approaches, may lead to better relaxation results. Nevertheless, for reasons of simplicity, here we stick with the described approximation scheme.

5.1.2. Matrix lifting revisited

In the last subsection, we described reduction schemes that serve the reduction of the dimension of SDP constraints. For this purpose, we exploited our knowledge about the eigenspace of B and tried to obtain more beneficial sets of eigenvalues $\{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n\}$ via approximations of the original parameter matrices. In consideration of the matrix lifting strategy, it is possible to exploit a low rank of B in quite the opposite way. In the following, we will describe a possibility to utilize the presence of a non-trivial nullspace of B, not for a reduction but for a tightening of the respective semidefiniteness condition.

We follow the index set definitions from the last subsection and utilize the compact eigenvalue decomposition $B = Q_{\Omega^0}\Lambda_{\Omega^0}Q_{\Omega^0}^T$ to establish the following identity:

$$\begin{bmatrix} I_{(|\Omega^0|)} & Q_{\Omega^0}^T X^T & Q_{\Omega^0}^T B X^T \\ XQ_{\Omega^0} & XQ_{\Omega^0}Q_{\Omega^0}^T X^T & XBX^T \\ XBQ_{\Omega^0} & XBX^T & XB^2X^T \end{bmatrix} = \begin{bmatrix} I_{(|\Omega^0|)} \\ XQ_{\Omega^0} \\ XBQ_{\Omega^0} \end{bmatrix} \begin{bmatrix} I_{(|\Omega^0|)} \\ XQ_{\Omega^0} \\ XBQ_{\Omega^0} \end{bmatrix}^T .$$

The semidefiniteness property of the left-hand side serves as a basis for a new relaxation framework. In this context, we first investigate the usability of the conditions

$$\begin{bmatrix} I_{(|\Omega^0|)} & Q_{\Omega^0}^T X^T & Q_{\Omega^0}^T B X^T \\ XQ_{\Omega^0} & G & Y \\ XBQ_{\Omega^0} & Y & Z \end{bmatrix} \in \mathcal{S}_+^{2n+|\Omega^0|} \tag{5.13a}$$

and

$$\begin{bmatrix} I_{(|\mho^0|)} & Q_{\mho^0}^T X^T \\ X Q_{\mho^0} & I - G \end{bmatrix} \in \mathcal{S}_+^{n+|\mho^0|}, \tag{5.13b}$$

as a replacement for the SDP constraint of ML_1.

If the rank of B is a good deal smaller than n, this substitution has only a slight effect on the overall computational effort. In rare cases, the reduced dimension of the matrix in (5.13a) can actually speed up the solving procedure. The more important matter is how this modification affects the quality of the matrix lifting based relaxation framework. The matrix variable G in (5.13a) and (5.13b) is used to relax the quadratic term $XQ_{\Omega^0}Q_{\Omega^0}^T X^T$. With the following result, we show that the presented substitution tightens the relaxation.

Lemma 5.1. *Any quadruple of matrices* (G, X, Y, Z) *that satisfies the semidefiniteness conditions in* (5.13a) *and* (5.13b) *also complies with constraint* (3.5b)*.*

Proof. Define the two block-diagonal matrices $D_{YZ} := \overset{*}{\operatorname{diag}}(Q_{\Omega^0}, I_{(2n)})$ and $D_G := \overset{*}{\operatorname{diag}}(Q_{\mho^0}, I_{(2n,n)})$. It is easy to check that the expression

$$D_{YZ} \begin{bmatrix} I_{(|\Omega^0|)} & Q_{\Omega^0}^T X^T & Q_{\Omega^0}^T B X^T \\ X Q_{\Omega^0} & G & Y \\ X B Q_{\Omega^0} & Y & Z \end{bmatrix} D_{YZ}^T + D_G \begin{bmatrix} I_{(|\mho^0|)} & Q_{\mho^0}^T X^T \\ X Q_{\mho^0} & I - G \end{bmatrix} D_G^T$$

is identical to the matrix in (3.5b). The positive definiteness of this matrix is a direct consequence of (5.13a) and (5.13b). □

Apparently, a smaller rank of matrix B decreases the trace of variable G. An decreased trace leads to a stronger SDP constraint which thereby improves the quality of the relaxation. For QAP instances with low-rank parameter matrices B, the improvement can be immense and is absolutely worth the slightly increased computational effort accompanied by the replacement of (4.31b) with (5.13).

One way to exploit this correlation beyond the already mentioned modifications is the utilization of a low-rank approximation $\tilde{B} = Q_{\Omega^\varepsilon} \Lambda_{\Omega^\varepsilon} Q_{\Omega^\varepsilon}$. Similarly to the approach used for QAP_{ESC}, however, the corresponding residual $R = B - \tilde{B}$ requires a special treatment. This includes the possible drawbacks accompanied by the selected approach. Although

the tighter bound for the significant term $\mathrm{tr}(AX\tilde{B}X^T)$ usually outweighs the possibly weaker relaxation of the remainder term, the improvements turn out to be relatively small, whereas the computational effort increases significantly. For now, we therefore dismiss the idea of constraint splittings based on low-rank approximation.

It is important to realize that the conclusion from above does by no means apply to the general idea of matrix splitting based rank reductions. By utilization of the Schur complement inequality to the respective matrix blocks, we see that (5.13a) involves the relation $Z \succeq YG^{\dagger}Y$. This inequality indicates a strong (nearly proportional) correlation between the expression $\|\!| G^{\frac{1}{4}}Z^{\frac{1}{2}}G^{\frac{1}{4}} - XBX^T |\!\|$ and the tightness of the semidefiniteness condition in (5.13a). By a loose interpretation of this connection, one may conclude that smaller values of $\mathrm{tr}(G)\,\mathrm{tr}(Z) = \|\!| BB^{\dagger} |\!\|_{\mathrm{F}}^2 \|\!| B |\!\|_{\mathrm{F}}^2$ indicate stronger relaxation bounds. In this regard, we are looking for a new splitting scheme $B = B_1 + B_2$ with the aim of minimizing the sum of the corresponding product terms $\sum_{i=1}^{2} \|\!| B_i B_i^{\dagger} |\!\|_{\mathrm{F}}^2 \|\!| B_i |\!\|_{\mathrm{F}}^2$. In order to achieve this, we apply a reverse optimization of the respective factors. This means that we concentrate on the minimization of the factors $\|\!| B_1 B_1^{\dagger} |\!\|_{\mathrm{F}}^2$ and $\|\!| B_2 |\!\|_{\mathrm{F}}^2$. In the actual implementation, we use a splitting scheme based on the spectral value decomposition of B. The splitting is realized in such a way that B_1 contains the most significant eigenvalues of B, but relatively few compared to the overall number of eigenvalues. The remainder part B_2, on the other hand, contains more eigenvalues with smaller absolute values. The individual application of the semidefiniteness condition in (5.13a) to each of these matrices results in a significant strengthening of the relaxation.

Other ideas for efficiency and quality improvements over the original relaxation framework QAP_{ML} are based on the reformulation and reduction strategies discussed in Section 4.1 as well as the approximation approach used in (5.6). We combine the addressed modifications and construct an extended SDP framework based on the matrix

lifting approach denoted QAP_{MLX} or simply *MLX*:

$$\inf_{X\in\mathcal{D}^n,\, G_i,Y_i,Z_i\in\mathcal{S}^n_+|_{i=1,2}} \quad \langle A, Y_1 + Y_2\rangle + \langle C, X\rangle \tag{5.14a}$$

$$\text{s. t.} \quad D_i^T \begin{bmatrix} I & X^T & BX^T \\ X & G_i & Y_i \\ XB & Y_i & Z_i \end{bmatrix} D_i \in \mathcal{S}_+, \qquad i = 1, 2, \tag{5.14b}$$

$$D_0^T \begin{bmatrix} I & X^T \\ X & I - G_1 - G_2 \end{bmatrix} D_0 \in \mathcal{S}_+, \tag{5.14c}$$

$$\begin{aligned} &\operatorname{diag}(Y_i) = X \operatorname{diag}(B_i), \quad \operatorname{diag}(Z_i) = X \operatorname{diag}(B_i^2), \\ &\operatorname{diag}(G_i) = X \operatorname{diag}(B_i B_i^\dagger), \qquad i = 1, 2, \end{aligned} \tag{5.14d}$$

$$\langle E, G_1 + G_2\rangle = \langle E, Z_1 + Z_2\rangle = 0, \tag{5.14e}$$

where the block-diagonal matrices $\{D_i\}$ are defined as $D_0 := \overset{*}{\operatorname{diag}}(Q_{\mathcal{U}^0}, I)$ and $D_i := \overset{*}{\operatorname{diag}}(Q_{\Omega_i^\varepsilon}, I, I)$ for $i \in \{1, 2\}$. The corresponding index sets $\{\Omega_i^\varepsilon\}$ are derived from the intersection of Ω^ε and the respective index sets which describe the eigenvalue assignment to the parts B_1 and B_2.

If the considered QAP instance requires a reduction of the computational expense, one may decrease the approximation threshold ε and replace the semidefinite constraints in (5.14c) by $I - G_1 - G_2 \in \mathcal{S}_+$. The complete removal of this constraint is generally not advisable. For many problem instances, the splitting scheme from above provides the opportunity for a further framework reduction. The described minimization procedure leads to a small rank of B_1. Often this matrix contains only a single nonzero eigenvalue or a single cluster of nonzero eigenvalues. If this is the case, we may replace D_1 with $\tilde{D}_1 := \overset{*}{\operatorname{diag}}(Q_{\Omega_1^\varepsilon}, I_{(2n,n)})$ and set $Y_1 = \frac{\operatorname{tr}(B_1)}{\operatorname{rank}(B_1)} G_1$. Even if there are two clusters of nonzero eigenvalues in B_1, it can be beneficial to replace the first constraint in (5.14b) with the respective semidefiniteness conditions based on the *ESC* approach.

5.1.3. Interrelated matrix splitting

A particularly beautiful property of the positive semidefinite matrix splitting scheme given in (3.9) is that the column spaces of the matrices B_+, B_- are orthogonal, such that $\mathcal{R}(B_+) \cap \mathcal{R}(B_-) = \{0\}$ and $B_+ B_- = B_- B_+ = 0_{(n,n)}$. As an immediate consequence, B_+ and B_- are moreover simultaneously diagonalizable. It would be a great advantage

if we could make use of these interrelations in the actual relaxation. Unfortunately, it seems quite difficult to exploit the corresponding properties in form of beneficial SDP constraints. For the design of new relaxation strategies, we need a different kind of interrelation.

In this subsection, we say goodbye to the idea of redundancy-free positive semidefinite matrix splitting pairs (B_+, B_-) and present a new splitting scheme:

$$B = B_\triangle - B_\triangledown \quad \text{with additional conditions on } (B_\triangle, B_\triangledown). \tag{5.15}$$

By the introduction of specific redundancies, we induce artificial correlations between the respective splitting parts. These interrelations shall be used to construct new types of constraints which are applicable in the corresponding SDP relaxation. To distinguish this new splitting from the non-redundant positive semidefinite (PSD) matrix splitting pair (B_+, B_-), we use the different notation in (5.15).

The possibilities of contrivable interrelations between $B_\triangle$ and $B_\triangledown$ are virtually endless. This, however, does by no means amount to a large number of properties that are suitable for our particular purpose. One of the few beneficial interrelation properties the author discovered in his research is the inverse semidefiniteness relation

$$B_\triangle \succeq B_\triangledown^{-1} \succeq 0. \tag{5.16}$$

The existence of the inverse $B_\triangledown^{-1}$ implies the regularity of $B_\triangledown$ and thereby also the regularity of $B_\triangle$. By the matrix equality

$$B_\triangledown - B_\triangle^{-1} = B_\triangle^{-1}(\underbrace{B_\triangle - B_\triangledown^{-1}}_{\succeq 0})B_\triangle^{-1} + (I - B_\triangledown^{-1}B_\triangle^{-1})^T \underbrace{B_\triangledown}_{\succeq 0}(I - B_\triangledown^{-1}B_\triangle^{-1}),$$

it is furthermore evident that (5.16) implies the validity of

$$B_\triangledown \succeq B_\triangle^{-1} \succeq 0 \tag{5.17}$$

Indeed, the conditions (5.16) and (5.17) are obviously equivalent.

The discussed interrelation property can be exploited by transferring the same to the relaxation variables for the quadratic terms $Y_\triangle = XB_\triangle X^T$ and $Y_\triangledown = XB_\triangledown X^T$. The

orthogonality of permutation matrices $X \in \Pi$ gives

$$XB_{\triangledown}^{-1}X^T = (XB_{\triangledown}X^T)^{-1}.$$

Relation (5.16) therefore requires $XB_{\triangle}X^T \succeq (XB_{\triangledown}X^T)^{-1} \succeq 0$, which provides the basis for the constraint $\mathsf{Y}_{\triangle} \succeq \mathsf{Y}_{\triangledown}^{-1} \succeq 0$. The latter condition can be realized by using the Schur complement inequality given in Theorem 2.1:

$$\begin{bmatrix} \mathsf{Y}_{\triangle} & I \\ I & \mathsf{Y}_{\triangledown} \end{bmatrix} \in \mathcal{S}_+^{2n}. \tag{5.18}$$

Before we can apply this constraint, we need to spend some thoughts on how to obtain a suitable matrix splitting pair $(B_{\triangle}, B_{\triangledown})$ satisfying the requirements from above. In the attempt of designing tight SDP relaxations, we are looking for matrix splitting pairs that accompany minimal norms.

Lemma 5.2. *For a symmetric $n \times n$ matrix $B \in \mathcal{S}^n$ and nonnegative coefficients $w_1, w_2 \in \mathbb{R}_+$ satisfying $w_1 + w_2 > 0$, consider the minimization problem*

$$\begin{aligned} \inf_{\mathsf{B}_{\triangle}, \mathsf{B}_{\triangledown} \in \mathcal{S}^n} \quad & w_1 |\!|\!| \mathsf{B}_{\triangle} |\!|\!| + w_2 |\!|\!| \mathsf{B}_{\triangledown} |\!|\!| \\ \text{s. t.} \quad & \mathsf{B}_{\triangle} \succeq \mathsf{B}_{\triangledown}^{-1} \succeq 0, \\ & \mathsf{B}_{\triangle} - \mathsf{B}_{\triangledown} = B. \end{aligned} \tag{5.19}$$

A solution to this program is given by the matrix pair $(\hat{B}_{\triangle}, \hat{B}_{\triangledown})$ defined as

$$\hat{B}_{\triangle} := \frac{1}{2}\left(B + \sqrt{B^2 + 4I}\right), \qquad \hat{B}_{\triangledown} := \hat{B}_{\triangle} - B. \tag{5.20}$$

This pair satisfies the identity $\hat{B}_{\triangle} = \hat{B}_{\triangledown}^{-1}$, and it is unique whenever the considered unitarily invariant matrix norms are strictly monotone.

Proof. The multiplication of both matrices gives

$$\hat{B}_{\triangle}\hat{B}_{\triangledown} = \frac{1}{2}\left(B + \sqrt{B^2 + 4I}\right)\frac{1}{2}\left(\sqrt{B^2 + 4I} - B\right) = \frac{1}{4}\left(B^2 + 4I - B^2\right) = I$$

and proves $\hat{B}_{\triangle} = \hat{B}_{\triangledown}^{-1}$. It is also straightforward to check that $(\hat{B}_{\triangle}, \hat{B}_{\triangledown})$ satisfies the constraints of problem (5.19). By regularity of $\mathsf{B}_{\triangle}, \mathsf{B}_{\triangledown}$ and the corresponding Schur complement inequality, we see that the semidefiniteness condition $\mathsf{B}_{\triangle} \succeq \mathsf{B}_{\triangledown}^{-1} \succeq 0$ is

equivalent to

$$\begin{bmatrix} B_\triangle & I \\ I & B_\triangledown \end{bmatrix} \in \mathcal{S}_+^{2n}.$$

Each solution vector $(B_\triangle, B_\triangledown)$ to (5.19) therefore corresponds to a matrix pair $(G, H) = (I, B_\triangle)$ that states a feasible point to the following programming problem

$$\begin{aligned} \inf_{G,H\in\mathcal{S}^n} \quad & w_1 |\!|\!| H |\!|\!| + w_2 |\!|\!| H - B |\!|\!| \\ \text{s. t.} \quad & \begin{bmatrix} H & G \\ G & H - B \end{bmatrix} \in \mathcal{S}_+^{2n}, \\ & q_i^T G q_i = 1 \qquad \text{for} \quad i \in \{k \mid \lambda_k \neq 0\}, \\ & G q_i = q_i \qquad \text{for} \quad i \in \{k \mid \lambda_k = 0\}. \end{aligned}$$

Any optimal point to this problem, which, conversely, relates to a matrix pair that is feasible in respect of problem (5.19), necessarily describes a solution to both. The remainder of the proof follows by Lemma 2.16. □

For the solution given in Corollary 5.2, one can explicitly state the introduced redundancy:

$$R = B_\triangle - B_+ = B_\triangledown - B_- = \frac{1}{2}\left(\sqrt{B^2 + 4I} - \sqrt{B^2}\right) \in \mathcal{S}_+^n,$$

with (B_+, B_-) taken from (3.9). The notion of redundancy defined in (3.1) is still a good indicator for the tightness of the considered relaxation. For that reasons, we will make use of this concept a little longer. Nevertheless, it should be taken into account that this term is used solely to demonstrate the difference to a splitting that is suitable for the relaxation program QAP_{MS}. It is also possible to adapt the notion of redundant and non-redundant positive semidefinite matrix splitting in respect of (5.18). In this context, the matrix pair $(B_\triangle, B_\triangledown)$ given in (5.20) would state a non-redundant splitting.

In Subsection 4.1.3, we claimed that none of the presented SDP relaxations is affected by a counterbalanced scaling of the parameter matrices A and B. However, the efficiency of (5.18) depends to a significant amount on the scaling of the data matrix B. For QAP

instances where $|||B|||_2$ is much greater than 1, the formulas in (3.8) and (5.20) give

$$\hat{B}_{\triangle} = \frac{1}{2}(B + \sqrt{B^2 + 4I}) \approx \frac{1}{2}(B + \sqrt{B^2}) = B_+, \quad \hat{B}_{\triangledown} \approx B_-.$$

Hence, in that case the splitting differs only slightly from the PSD splitting based on the spectral value decomposition, and the effect of the inverse interrelation on the corresponding feasible set is hardly noticeable. On the other hand, if $|||B|||_2 \ll 1$, the validity of (5.16) is purchased by introducing a relatively large redundancy:

$$\hat{B}_{\triangle} = \frac{1}{2}(B + \sqrt{B^2 + 4I}) \approx I, \quad \hat{B}_{\triangledown} \approx I \quad \Longrightarrow \quad |||R|||_2 \approx 1.$$

To counteract this behavior, we apply a linear homogeneous function $\tau\colon \mathcal{M}^n \to \mathbb{R}$ and replace (5.16) with

$$B_{\triangle} \succeq \tau(B)^2 B_{\triangledown}^{-1} \succeq 0. \tag{5.21}$$

For any positive real scaling factor α, the condition

$$\alpha B_{\triangle} \succeq \tau(\alpha B)^2 (\alpha B_{\triangledown})^{-1} \succeq 0$$

is equivalent to (5.21). The equivalence is easily apparent from the linearity of τ and demonstrates scaling invariance of this relation. In consideration of the semidefiniteness condition and the applicable equality constraints, we suggest the trace norm of a projection of B as a suitable base for τ. In the actual implementation, we use the renormalization function τ defined as

$$\tau(B) := \frac{3}{5n} |||PBP|||_{\mathrm{T}}, \tag{5.22}$$

where the orthogonal projection matrix P is defined as $P := I - \frac{1}{n}E$. Among the tested matrix norms and various scalings of these, the particular choice given in (5.22) worked best for a large range of problems.

Regarding the nullspace of B - particularly its exploitation by the respective SDP constraints -, another striking circumstance becomes apparent. In the last two subsections, we demonstrated strategies for the utilization of a possible low rank of B. The existence of some non-trivial nullspace was used to either improve the quality or reduce the computational costs of the corresponding relaxation frameworks. However, in consideration of constraint (5.18), a low rank of B is rather ballast than a beneficial

property. This issue can be resolved by replacing the inverse property in (5.16) with the pseudoinverse relations

$$B_{\triangle} \succeq B_{\triangledown}^{\dagger} \succeq 0 \qquad \text{and} \qquad B_{\triangledown} \succeq B_{\triangle}^{\dagger} \succeq 0. \tag{5.23}$$

By Lemma 2.13, it is apparent that any matrix pair $(B_{\triangle}, B_{\triangledown})$ that complies with these two conditions necessarily satisfies

$$\mathcal{R}(B_{\triangle}) \supseteq \mathcal{R}(B_{\triangledown}^{\dagger}) = \mathcal{R}(B_{\triangledown}) \supseteq \mathcal{R}(B_{\triangle}^{\dagger}) = \mathcal{R}(B_{\triangle}),$$

such that $\mathcal{R}(B_{\triangle}) = \mathcal{R}(B_{\triangledown})$. This, in turn, demonstrates the equivalence of (5.23) and the condition

$$\begin{bmatrix} B_{\triangle} & G \\ G & B_{\triangledown} \end{bmatrix} \in \mathcal{S}_{+}^{2n},$$

where G is the orthogonal projection matrix for the space $\mathcal{R}(B_{\triangle}) \cup \mathcal{R}(B_{\triangledown})$.

For the actual implementation, we take the approach one step further by incorporating the renormalization function τ and weighting the utilization of the inverse interrelation property against the introduced redundancy. In order to achieve these objectives, we apply the following program:

$$\begin{aligned} \inf_{\mathbf{B}_{\triangle}, \mathbf{B}_{\triangledown}, \mathbf{G} \in \mathcal{S}^n} \quad & \operatorname{tr}(w_1 \mathbf{B}_{\triangle} + w_2 \mathbf{B}_{\triangledown} - \xi \mathbf{G}) \\ \text{s. t.} \quad & \begin{bmatrix} \mathbf{B}_{\triangle} & \mathbf{G} \\ \mathbf{G} & \mathbf{B}_{\triangledown} \end{bmatrix} \in \mathcal{S}_{+}^{2n}, \\ & \mathbf{B}_{\triangle} - \mathbf{B}_{\triangledown} = B, \\ & \|\!| \mathbf{G} |\!\|_2 \leqslant \tau(B). \end{aligned} \tag{5.24}$$

The solution to this program can be determined analytically.

Theorem 5.3. *For given $B \in \mathcal{S}^n$ with eigenvalue decomposition $B = \sum_{i=1}^n \lambda_i q_i q_i^T$, coefficients $w_1, w_2 \in \mathbb{R}_+$ satisfying $w_1 + w_2 = 2$, and a real value $\xi \in [0, 2)$, define*

$$\begin{aligned} G &:= \sum_{i=1}^n \min\{\tau(B), \frac{\xi|\lambda_i|}{2\sqrt{4-\xi^2}}\} q_i q_i^T, \\ B_\triangle &:= \frac{1}{2}B + \sqrt{\frac{1}{4}B^2 + G^2}, \\ B_\triangledown &:= B_\triangle - B. \end{aligned} \tag{5.25}$$

The matrix triple $(B_\triangle, B_\triangledown, G)$ states the unique solution to problem (5.24).

Proof. Denote by $(\hat{B}_\triangle, \hat{B}_\triangledown, \hat{G})$ an optimal point to the considered minimization problem, define $\zeta := [q_1^T \hat{G} q_1, \ldots, q_n^T \hat{G} q_n]$, and consider

$$\begin{aligned} \inf_{B_\triangle, B_\triangledown, \in \mathcal{S}^n_+,\, G \in \mathcal{M}^n} \quad & w_1 \|\!| B_\triangle |\!\|_{\mathrm{T}} + w_2 \|\!| B_\triangledown |\!\|_{\mathrm{T}} - \xi \langle e, \zeta \rangle \\ \text{s. t.} \quad & \begin{bmatrix} B_\triangle & G \\ G & B_\triangledown \end{bmatrix} \in \mathcal{S}^{2n}_+, \\ & q_i^T G q_i = \zeta_i \qquad \text{for} \quad i = 1, \ldots, n, \\ & B_\triangle - B_\triangledown = B. \end{aligned} \tag{5.26}$$

Apparently, any solution to this problem is also a solution to the minimization problem (5.24). By Lemma 2.16, it follows the existence of an optimal point $(\hat{B}_\triangle, \hat{B}_\triangledown, \hat{G})$ with all three matrices being diagonalizable by the same eigenvectors $\{q_i\}$. In addition, Lemma 2.16 validates the explicit formula for $\hat{B}_\triangle$. Inserting this formula into the objective function gives

$$\begin{aligned} \operatorname{tr}(w_1 \hat{B}_\triangle + w_2 \hat{B}_\triangledown) - \xi \operatorname{tr}(\hat{G}) &= \operatorname{tr}(2\hat{B}_\triangle - w_2 B) - \xi \operatorname{tr}(\hat{G}) \\ &= \operatorname{tr}\left(\sqrt{B^2 + 4\hat{G}^2}\right) + (1 - w_2)\operatorname{tr}(B) - \xi \operatorname{tr}(\hat{G}) \\ &= \sum_{i=1}^n q_i^T \left(\sqrt{B^2 + 4\hat{G}^2} + (1 - w_2)B\right) q_i - \xi q_i^T \hat{G} q_i \\ &= \sum_{i=1}^n \sqrt{\lambda_i^2 + 4 q_i^T \hat{G}^2 q_i} + (1 - w_2)\lambda_i - \xi q_i^T \hat{G} q_i \\ &= \sum_{i=1}^n \sqrt{\lambda_i^2 + 4\zeta_i^2} + (1 - w_2)\lambda_i - \xi \zeta_i =: \mathrm{h}(\zeta). \end{aligned}$$

Since $\mathrm{h}(\zeta)$ is a sum of univariate functions, its minimization can be done element-wise for each summand separately. We define the anticipated minimizer as

$$\hat{\zeta} := \frac{\xi}{2\sqrt{4-\xi^2}}|\lambda(B)|. \tag{5.27}$$

From the gradient of h

$$\nabla \mathrm{h}(\zeta) = \left(\frac{4\zeta_i}{\sqrt{\lambda_i^2 + 4\zeta_i^2}} - \xi \right)$$

it is clear that this function is element-wise strictly monotonically decreasing in the interval $(-\infty, \hat{\zeta})$ and element-wise strictly monotonically increasing in the interval $(\hat{\zeta}, \infty)$.

Additionally, the spectral norm inequality $\|\!|\hat{G}|\!\|_2 \leqslant \tau(B)$ implies the same bounds on the variables $\zeta_i = q_i^T \hat{G} q_i \leqslant \tau(B)$. This leads to the following adjustment of (5.27):

$$\hat{\zeta} := (\hat{\zeta}_i) \qquad \text{with} \quad \hat{\zeta}_i = \min\{\tau(B), \frac{\xi}{2\sqrt{4-\xi^2}}|\lambda_i|\}. \tag{5.28}$$

Though problem (5.26) does not necessarily accompany a unique solution, the uniqueness of $\hat{\zeta}$ is evident due to the continuity of h and its monotonicity properties. Since the formula in (5.28) implies that the set $\{i \mid \lambda_i = 0, \hat{\zeta}_i \neq 0\}$ is empty, all requirements in Lemma 2.16 for a unique solution are met. □

The parameter ξ serves as a threshold for the introduced redundancy and, in a certain way, also as a threshold for the effectiveness of the generalized inverse interrelation. For the extremes $\xi = 0$ and $\xi > 2$, the respective semidefiniteness conditions fall back to the SDP constraints used in the pure non-redundant matrix splitting approach from [83] and the full normalized inverse property given in (5.21), respectively. By no means, however, ξ is used as a trade-off between speed and quality of the respective relaxations. The best bounding results are obtained for values in between these extremes. For the numerical examples in the following sections, we use $\xi = \frac{3}{2}$ since this value works well for a large range of problems.

Depending on the particular instance, it can be advantageous to utilize an individual QAP reformulation that is more optimized for the discussed interrelated matrix splitting approach. For this purpose, one simply needs to modify problem (5.24) by replacing its equality condition with $B_\triangle - B_\triangledown = B + \mathrm{diag}^*(d_B) + v_B e^T + e v_B^T$. From the solution of this problem, one derives new reformulation parameters $\hat{d}_B$ and $\hat{v}_B$ which are specifically

designed in consideration of this new type of SDP constraints. It can be shown that the sum-matrix vector $\hat{v}_B$ again satisfies the formula in (4.9). On the other hand, the diagonal vector $\hat{d}_B$ usually differs from its counterpart in (4.19). Nevertheless, since the differences are typically relatively small and for reasons of comparability, we are using the QAP reformulation (4.20) for all upcoming numerical examples.

The last piece in the puzzle of designing a new matrix splitting based SDP relaxation for the QAP is the construction of the corresponding quadratic semidefiniteness conditions. For the optimal matrix triple defined in (5.25), we have $G = B_\triangle^{\frac{1}{2}} B_\triangledown^{\frac{1}{2}} = B_\triangledown^{\frac{1}{2}} B_\triangle^{\frac{1}{2}}$. Together with the conic inequality for the ε-approximate decomposition - by which we refer to $B \succeq Q_{\Omega^\varepsilon} \Lambda_{\Omega^\varepsilon} Q_{\Omega^\varepsilon}^T$ -, one obtains

$$\begin{bmatrix} B_\triangle & G \\ G & B_\triangledown \end{bmatrix} \geq \begin{bmatrix} B_\triangle^{\frac{1}{2}} Q_{\Omega^\varepsilon} \\ B_\triangledown^{\frac{1}{2}} Q_{\Omega^\varepsilon} \end{bmatrix} \begin{bmatrix} B_\triangle^{\frac{1}{2}} Q_{\Omega^\varepsilon} \\ B_\triangledown^{\frac{1}{2}} Q_{\Omega^\varepsilon} \end{bmatrix}^T.$$

In the following relaxation framework, this condition is implemented by using the Schur complement inequality. To that end, we further define the diagonal matrix $D_\tau := \tau(B)I$ as well as the block diagonal matrices $D_Y := \operatorname{diag}^*(Q_{\Omega^\varepsilon}, I_{(2n)})$ and $D_G := \operatorname{diag}^*(Q_{\mathfrak{V}_G^{1-\varepsilon}}, I_{(n)})$, where $\mathfrak{V}_G^\varepsilon := \{i \mid |\lambda_i(G)| < \varepsilon \|\!| G |\!\|_2\}$ denotes an adaptation of the index set definitions in (5.4) for the eigenvalues of G.

Finally, we are in the position to present the level-1 version of the inverse interrelated matrix splitting relaxation (*IIMS*):

$$\begin{aligned}
\inf_{X \in \mathcal{D}^n,\ G,Y,Y_\triangle,Y_\triangledown \in \mathcal{S}^n} \quad & \langle A, Y \rangle + \langle C, X \rangle && \text{(5.29a)} \\
\text{s. t.} \quad & D_Y^T \begin{bmatrix} I & B_\triangle^{\frac{1}{2}} X^T & B_\triangledown^{\frac{1}{2}} X^T \\ X B_\triangle^{\frac{1}{2}} & Y_\triangle & G \\ X B_\triangledown^{\frac{1}{2}} & G & Y_\triangledown \end{bmatrix} D_Y \in \mathcal{S}_+^{2n+|\Omega^\varepsilon|}, && \text{(5.29b)} \\
& D_G^T \begin{bmatrix} \left(D_\tau - B_\triangle^{\frac{1}{2}} B_\triangledown^{\frac{1}{2}}\right)^\dagger & X^T \\ X & D_\tau - G \end{bmatrix} D_G \in \mathcal{S}_+^{n+|\mathfrak{V}_G^{1-\varepsilon}|}, && \text{(5.29c)} \\
& \begin{aligned} &\operatorname{diag}(Y_\triangle) = X \operatorname{diag}(B_\triangle), \quad \operatorname{diag}(Y_\triangledown) = X \operatorname{diag}(B_\triangledown), \\ &\operatorname{diag}(G) = X \operatorname{diag}(B_\triangle^{\frac{1}{2}} B_\triangledown^{\frac{1}{2}}), \end{aligned} && \text{(5.29d)} \\
& \langle Y_\triangle, E \rangle = \langle B_\triangle, E \rangle, \quad \langle Y_\triangledown, E \rangle = \langle B_\triangledown, E \rangle, && \text{(5.29e)} \\
& Y = Y_\triangle - Y_\triangledown. && \text{(5.29f)}
\end{aligned}$$

Table 5.1.: Selected bounds for comparison of framework modifications [$R_{\text{gap}} = 1 - \frac{\text{Relaxation result}}{\text{Upper bound}}$ in (%)]

Problem	ML_1	MLX_1	MS_1	$IIMS_1$	ES_1	ESC_1
Esc32g	1028.26	**566.67**	**566.67**	**566.67**	**566.67**	**566.67**
Kra32	35.05	18.22	26.25	21.76	**14.24**	14.27
LiPa40a	2.27	2.20	4.20	2.37	**2.10**	**2.10**
Nug30	29.25	8.65	12.14	8.65	**7.99**	8.02
Ste36a	135.37	32.56	49.12	30.55	**26.85**	26.87
Tai35a	19.16	18.79	30.50	19.80	**17.90**	**17.90**
Tho40	53.85	13.26	14.54	12.58	**11.42**	11.43

Using the Schur complement inequality to obtain constraint (5.29c) has usually only a small effect on the quality of the computed bounds. For reasons of efficiency, it may therefore be beneficial to refrain from the incorporation of the full constraint and replace it with the simplified semidefiniteness condition $D_\tau - G \geq 0$.

5.2. Intermediate comparison of level-1 relaxations

In consonance with the general structure of this thesis, we conclude this chapter with a small reflection on the discussed modifications. For this purpose, we compare bounds obtained by solving the new relaxation programs MLX_1, $IIMS_1$, and ESC_1 with the results attained by applying their origins ML_1, MS_1, and ES_1, respectively. Reasoned in the circumstance that we introduced no alternative relaxation framework based on the vector lifting technique, we omit the program VL_1 from the following consideration. The absence of this relaxation framework enables us to compute lower bounds for QAP instances of greater sizes. For the following numerical results, the author chose different problems with dimensions n between 30 and 40. Nevertheless, it should be kept in mind, that VL_1 remains to be the strongest relaxation presented yet.

The bounds in Table 5.1 are computed using $\varepsilon = 0$. The setup of all other parameters follows the suggestions for the corresponding programming problem definitions, (4.31), (4.32), (4.33), (5.14), (5.29), and (5.11). The numerical results demonstrate the enhance-

Table 5.2.: Bound computations for $\varepsilon = 0.2$

Prob.	MLX_1	$IIMS_1$	ESC_1
Esc32g	**566.67**	**566.67**	**566.67**
Kra32	18.22	21.76	**14.27**
LiPa40a	**2.24**	2.39	3.24
Nug30	**8.80**	8.81	9.97
Ste36a	35.08	**31.96**	38.70
Tai35a	**18.91**	19.83	24.03
Tho40	14.48	**13.75**	14.64

ment of the frameworks *MLX* and *IIMS* over their origins *ML* and *MS*, respectively. Overall, the matrix lifting based relaxation performs superior in comparison to the programs based on matrix splitting, but there is still no ordering between these approaches. Table 5.1 also demonstrates the very small effect of the "eigenspace clustering" on the relaxation quality of *ES*. The parameter matrices of the QAP instances *Esc32g*, *Kra32*, and *Ste36a* involve 2, 5, and 12 clusters of eigenvalues, respectively. This leads to a significant reduction of the associated instances of *ESC* with a hardly noticeable effect on the computed bounds.

Another selection of bounds, given in Table 5.2, shall be used to illustrate the influence of constraint approximations. The bounds are computed for a relative approximation tolerance of 20%, i.e. $\varepsilon = 0.2$. Additionally, we limited the maximal number of semidefinite matrices in the respective ESC_1 instances to 10. In the presence of the residual variables F_+ and F_- this usually implies that B is approximated by $\tilde{B}$ with no more than 8 clusters of eigenvalues. The problem instance *Nug30* is the only one for which this limitation leads to a slight violation of the relative approximation tolerance.

The application of the described approximation procedure has little impact on the relaxation quality of the frameworks MLX_1 and $IIMS_1$. The bounds are marginally weaker than their counterparts in Table 5.1. On the other hand, the approximation has a significant influence on the quality of ESC_1. Even in relation to the considerable reduction of the computational effort, the weakening of ESC_1 is drastic. For the majority

of the tested QAP instances, the "eigenspace clustering" SDP relaxation is actually outperformed by the less expensive frameworks MLX_1 and $IIMS_1$.

The most important observation from the presented numerical results is that the quality of the bounds obtained via the new frameworks MLX_1 and $IIMS_1$ is very close to the bounding quality of the relaxation ES_1. In regard to the lower computational complexity accompanied by the former, this circumstance is very promising. Nevertheless, the framework ES_1 remains to have a superior bounding quality in comparison to the low-dimensional relaxations. Other tests have also shown that the quality of the "eigenspace" SDP relaxation is less sensitive to inappropriate choices of QAP relaxations.

As in Theorem 3.2, it is possible to prove similar ordering properties for the newly introduced relaxation frameworks. The following corollary shows that the rough connection between the complexity and the quality of the respective relaxation still holds true.

Corollary 5.4. *For a QAP instance* (A, B, C)*, denote by* $\hat{\varrho}_{\mathrm{ML}_1}$*,* $\hat{\varrho}_{\mathrm{MS}_1}$*,* $\hat{\varrho}_{\mathrm{ES}_1}$*,* $\hat{\varrho}_{\mathrm{MLX}_1}$*,* $\hat{\varrho}_{\mathrm{IIMS}_1}$*, and* $\hat{\varrho}_{\mathrm{ESC}_1}$ *the optimal objective values to the problems* (4.31)*,* (4.32)*,* (4.33)*,* (5.14)*,* (5.29)*, and* (5.11)*, respectively. They satisfy the relation*

$$\hat{\varrho}_{\mathrm{ES}_1} \geqslant \max\{\hat{\varrho}_{\mathrm{ML}_1}, \hat{\varrho}_{\mathrm{MLX}_1}, \hat{\varrho}_{\mathrm{MS}_1}, \hat{\varrho}_{\mathrm{IIMS}_1}\}. \tag{5.30a}$$

Moreover, if the approximation tolerance ε *is set to zero, then*

$$\hat{\varrho}_{\mathrm{MLX}_1} \geqslant \hat{\varrho}_{\mathrm{ML}_1} \tag{5.30b}$$

and

$$\hat{\varrho}_{\mathrm{ES}_1} \geqslant \hat{\varrho}_{\mathrm{ESC}_1} \geqslant \max\{\hat{\varrho}_{\mathrm{ML}_1}, \hat{\varrho}_{\mathrm{MS}_1}, \hat{\varrho}_{\mathrm{IIMS}_1}\}. \tag{5.30c}$$

Proof. We start with the argument for (5.30b) and (5.30c), thus assume $\varepsilon = 0$. Inequality (5.30b) is an immediate consequence of Lemma 5.1 and the fact that the additional constraints on G as well as the applied splitting procedure only tighten the relaxation. Similarly evident is the relation $\hat{\varrho}_{\mathrm{ES}_1} \geqslant \hat{\varrho}_{\mathrm{ESC}_1}$. For a solution vector $(\hat{X}, \hat{Q}_1, \ldots, \hat{Q}_n, \hat{Y})$ to problem (4.33), it is possible to construct another vector

$$(X, F_+, F_-, U_1, \ldots, U_k, Y) := \left(\hat{X}, 0_{(n,n)}, 0_{(n,n)}, \sum_{i \in \Phi_1} \hat{Q}_i, \ldots, \sum_{i \in \Phi_k} \hat{Q}_i, \hat{Y}\right),$$

that states a feasible point to problem (5.11) and accompanies the same objective value.

For the remainder of the first argument, let $(\hat{X}, \hat{F}_+, \hat{F}_-, \hat{U}_1, \ldots, \hat{U}_k, \hat{Y})$ denote an optimal point to the respective instance of framework ESC_1. By $\varepsilon = 0$, we have $R_+ = R_- \equiv 0$ which also necessitates $\hat{F}_+ = \hat{F}_- \equiv 0$. The subsequent argument follows the same approach used a number of times before. By construction, we show the existence of feasible points of the weaker relaxations whose objective values are identical to $\hat{\varrho}_{\mathrm{ESC}_1}$. In consideration of the relaxations (4.32) and (4.31), these points are

$$(X, Y, Y_+, Y_-) := \left(\hat{X}, \hat{Y}, \sum_{i\colon \lambda_{i\star}>0} \lambda_{i\star}\hat{U}_i, \sum_{i\colon \lambda_{i\star}<0} -\lambda_{i\star}\hat{U}_i\right)$$

and

$$(X, Y, Z) := \left(\hat{X}, \hat{Y}, \sum_{i=1}^{k} \lambda_{i\star}^2 \hat{U}_i\right),$$

respectively. The feasibility of the former is easily shown and, due to $(X, Y) = (\hat{X}, \hat{Y})$, the equivalence of the corresponding objective values is evident. It remains to show that (X, Y, Z) satisfies the other constraints of the relaxation ML_1.

Apparently, the diagonal equalities in (5.11d) necessitate the validity of the respective equality conditions in (4.32c). By using Lemma 4.8, it is similarly straightforward to show the validity of the constraints in (4.31d). Furthermore, the identities $\sum_{i=1}^{k} \hat{U}_i = I = \sum_{i=1}^{k} Q_{\Phi_i} Q_{\Phi_i}^T$ imply the following relation:

$$\begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} = \sum_{i=0}^{k} \begin{bmatrix} Q_{\Phi_i}^T & 0 \\ 0 & I \\ 0 & \lambda_{i\star} I \end{bmatrix} \underbrace{\begin{bmatrix} I_{(|\Phi_i|)} & Q_{\Phi_i}^T \hat{X}^T \\ \hat{X} Q_{\Phi_i} & \hat{U}_i \end{bmatrix}}_{\succeq 0} \begin{bmatrix} Q_{\Phi_i}^T & 0 \\ 0 & I \\ 0 & \lambda_{i\star} I \end{bmatrix}^T .$$

The nonnegative definiteness conditions in (5.11b) therefore require the compliance of (X, Y, Z) with constraint (4.31b). Taken together, this validates $\hat{\varrho}_{\mathrm{ESC}_1} \geqslant \hat{\varrho}_{\mathrm{ML}_1}$.

In order to show $\hat{\varrho}_{\mathrm{ESC}_1} \geqslant \hat{\varrho}_{\mathrm{IIMS}_1}$, we define the functions

$$\begin{aligned} \mathrm{g}_G(\lambda_i) &:= \min\{\tau(B), \frac{\xi}{2\sqrt{4-\xi^2}}|\lambda_i|\}, \\ \mathrm{g}_\triangle(\lambda_i) &:= \frac{\lambda_i}{2} + \frac{1}{2}\sqrt{\lambda_i^2 + 4\mathrm{g}_G(\lambda_i)}, \\ \mathrm{g}_\triangledown(\lambda_i) &:= \mathrm{g}_\triangle(\lambda_i) - \lambda_i, \\ \mathrm{g}_\tau(\lambda_i) &:= \tau(B) - \mathrm{g}_G(\lambda_i), \end{aligned}$$

where the coefficient ξ and the function τ are the same as in problem (5.29). These functions are used for the construction of the point

$$(G, X, Y, Y_{\triangle}, Y_{\triangledown}) := \left(\sum_{i=1}^{k} \mathrm{g}_G(\lambda_{i\star}) \hat{U}_i,\ \hat{X},\ \hat{Y},\ \sum_{i=1}^{k} \mathrm{g}_{\triangle}(\lambda_{i\star}) \hat{U}_i,\ \sum_{i=1}^{k} \mathrm{g}_{\triangledown}(\lambda_{i\star}) \hat{U}_i \right).$$

Once more, the equivalence of the accompanied objective values as well as the compliance with the equality constraints of $IIMS_1$ are evident. The validity of the corresponding semidefiniteness conditions is proved by observing that the sums of positive semidefinite matrices

$$\sum_{i=1}^{k} \begin{bmatrix} Q_{\Omega^0}^T Q_{\Phi_i} & 0 \\ 0 & \sqrt{\mathrm{g}_{\triangle}(\lambda_{i\star})} I \\ 0 & \sqrt{\mathrm{g}_{\triangledown}(\lambda_{i\star})} I \end{bmatrix} \underbrace{\begin{bmatrix} I_{(|\Phi_i|)} & Q_{\Phi_i}^T \hat{X}^T \\ \hat{X} Q_{\Phi_i} & \hat{U}_i \end{bmatrix}}_{\succeq 0} \begin{bmatrix} Q_{\Omega^0}^T Q_{\Phi_i} & 0 \\ 0 & \sqrt{\mathrm{g}_{\triangle}(\lambda_{i\star})} I \\ 0 & \sqrt{\mathrm{g}_{\triangledown}(\lambda_{i\star})} I \end{bmatrix}^T$$

and

$$\sum_{i=1}^{k} \underbrace{\mathrm{g}_{\tau}(\lambda_{i\star})^{\dagger}}_{\geqslant 0} \begin{bmatrix} Q_{\mho^1}^T Q_{\Phi_i} & 0 \\ 0 & \mathrm{g}_{\tau}(\lambda_{i\star}) I \end{bmatrix} \underbrace{\begin{bmatrix} I_{(|\Phi_i|)} & Q_{\Phi_i}^T \hat{X}^T \\ \hat{X} Q_{\Phi_i} & \hat{U}_i \end{bmatrix}}_{\succeq 0} \begin{bmatrix} Q_{\mho^1}^T Q_{\Phi_i} & 0 \\ 0 & \mathrm{g}_{\tau}(\lambda_{i\star}) I \end{bmatrix}^T$$

are identical to the respective matrices in (5.29b) and (5.29c).

It remains to prove the validity of the relations in (5.30a). Actually, it is sufficient to prove the inequality $\hat{\varrho}_{\mathrm{ES}_1} \geqslant \hat{\varrho}_{\mathrm{MLX}_1}$ since the validity of the other relations is an immediate consequence of (5.30c) and the observation that positive approximation tolerances $\varepsilon > 0$ can only weaken the relaxations whose optimal objective values are listed on the right-hand side of (5.30c). On the other hand, the relaxation ES_1 is not affected by the parameter ε. For the proof of $\hat{\varrho}_{\mathrm{ES}_1} \geqslant \hat{\varrho}_{\mathrm{MLX}_1}$, we assume that $(\hat{X}, \hat{Q}_1, \ldots, \hat{Q}_n, \hat{Y})$ denotes an optimal point to problem (4.33). It is then possible to construct a feasible point $(G_1, G_2, \hat{X}, Y_1, Y_2, Z_1, Z_2)$ to problem (5.14) that accompanies the same objective value $\hat{\varrho}_{\mathrm{ES}_1}$. The construction rule for this point is

$$(G_j, Y_j, Z_j) := \left(\sum_{i \in \Omega_j^0} \hat{Q}_i,\ \sum_{i \in \Omega_j^0} \lambda_i \hat{Q}_i,\ \sum_{i \in \Omega_j^0} \lambda_i^2 \hat{Q}_i \right) \qquad \text{for} \quad j = 1, 2.$$

The compliance with the equality constraints of MLX_1 follows immediately from their counterparts in relaxation ES_1 and the definition from above.

By the identities $B_j Q_{\Omega_j^0} = Q_{\Omega_j^0}\Lambda_{\Omega_j^0} = \sum_{i\in\Omega_j^0}\lambda_i q_i q_i^T Q_{\Omega_j^0}$ and the respective equality constraints in (4.33), it can be shown that

$$\sum_{i\in\Omega_j^0}\begin{bmatrix} Q_{\Omega_j^0}^T q_i & 0 \\ 0 & I \\ 0 & \lambda_i I \end{bmatrix}\begin{bmatrix} 1 & q_i^T\hat{X}^T \\ \hat{X}q_i & \hat{Q}_i \end{bmatrix}\begin{bmatrix} Q_{\Omega_j^0}^T q_i & 0 \\ 0 & I \\ 0 & \lambda_i I \end{bmatrix}^T = \begin{bmatrix} I_{(|\Omega_j^0|)} & Q_{\Omega_j^0}^T X^T & Q_{\Omega_j^0}^T B X^T \\ X Q_{\Omega_j^0} & G_j & Y_j \\ X B Q_{\Omega_j^0} & Y_j & Z_j \end{bmatrix}$$

for each index $j \in \{1,2\}$. By elimination of the row and column vectors specified in $\Omega_j^0\backslash\Omega_j^\varepsilon$, we derive block matrices which match the semidefiniteness conditions in (5.14b). The validation of the compliance with (5.14c) follows a very similar procedure. □

From the proof of Corollary 5.4, it is obvious that the validity of the inequalities in (5.30) is not limited to a particular QAP reformulation, not even a fixed reformulation for all considered relaxations. The necessary conditions are that the parameter matrices $B, B_+, B_\triangle, G$ are all simultaneously diagonalizable and that the eigenvalue clusters of B with the accompanied invariant subspaces have their analogs within the matrices $B_+, B_\triangle$, and G.

If we assume that the matrix splitting scheme for relaxation *MLX* does not separate any two eigenvalues from the same cluster of eigenvalues, then also the relation $\hat{\varrho}_{\mathrm{ESC}_1} \geqslant \hat{\varrho}_{\mathrm{MLX}_1}$ is provably correct. The argument is similar to that one used for Theorem 3.2 and Corollary 5.4.

Naturally, the result given in Lemma 3.3 can be extended in regard to the newly introduced relaxation frameworks.

Corollary 5.5. *For a given QAP instance, let* $(\hat{X},\ldots)$ *denote a solution vector to one of the relaxation problems* (4.29), (4.31), (4.32), (4.33), (5.11), (5.14), *or* (5.29). *If the approximation tolerance* ε *is set to zero and* $\hat{X} \in \Pi$, *then* $\hat{X}$ *describes an optimal assignment for the actual QAP. The corresponding optimal objective values are identical.*

Proof. The proof for each SDP relaxation is similar to that of Lemma 3.3. □

Chapter 6.

Cutting Strategies

6.1. Cuts

Besides the already mentioned reformulation approaches, a common instrument to tighten a relaxation is the incorporation of additional non-redundant constraints. In the following subsections, we elaborate a few strategies to derive such constraints.

6.1.1. Gilmore-Lawler bound constraints

The Gilmore-Lawler bound (*GLB*) [37,60] is one of the most famous bounding techniques for the quadratic assignment problem. Its mainstream awareness is not only reasoned in the simplicity of the *GLB* but also in its good performance. For an instance (A, B, C) of (KBQAP) the Gilmore-Lawler bound relaxation is given by the following linear assignment problem (LAP):

$$\min_{X \in \Pi^n} \quad \langle L + C, X \rangle, \tag{6.1a}$$

where

$$L := (l_{ij}) \qquad \text{with} \quad l_{ij} = \min_{X \in \Pi^n,\, X_{ij}=1} (AXB)_{ij}. \tag{6.1b}$$

The computation of the coefficients (l_{ij}) as permuted dot products reduces the overall complexity of the *GLB* to $\mathcal{O}(n^3)$, see [37]. Its low computational cost and the comparatively good bounds are stimuli for us to incorporate the *GLB* into the considered SDP relaxations.

By definition of L, we easily see that for each $X \in \Pi^n$:

$$AXBX^T \geqslant_{\text{diag}} LX^T. \tag{6.2}$$

This equation provides the opportunity to incorporate n additional linear constraints into the respective relaxation frameworks. The integration can be realized simply by adding the inequality condition

$$AY \geqslant_{\text{diag}} LX^T. \tag{6.3}$$

Relaxations based on the vector lifting technique allow a deeper integration of the *GLB* conditions. By $(B^T \otimes A)\,\text{vec}(X) = \text{vec}(AXB)$, we derive the identity

$$\text{diag}((B^T \otimes A)\,\text{vec}(X)\,\text{vec}(X)^T) = \text{vec}((AXB) \circ X),$$

which, in turn, gives

$$\text{diag}((B^T \otimes A)\Upsilon) \geqslant \text{vec}(L \circ X). \tag{6.4}$$

This vector inequality utilizes every single product of the term $L \circ X$, whereas the identity $\text{diag}(LX^T) = (L \circ X)e$ shows that (6.3) exploits solely the sums of the respective rows. Apparently, (6.4) implies the validity of (6.3), thus leads to tighter relaxations. Since the additional computational costs are small compared to the overall efforts of VL_1, we suggest to use (6.4).

By incorporating the *GLB* based constraints into the respective SDP relaxations, we evidently obtain stronger bounds than with the plain Gilmore-Lawler bound procedure. More specifically, by using Lemma 3.3 and Corollary 5.5, we show the following result.

Corollary 6.1. *In respect of a given QAP instance* (A, B, C)*, consider any of the previously discussed level-*1 *relaxations, and additionally incorporate condition* (6.3)*. The optimal objective value to this SDP relaxation is always greater than or equal to the optimal objective value to problem* (6.1)*. Moreover, if the approximation tolerance* ε *is zero, the solution vector to the respective instance of problem* (6.1) *is unique, and the optimal objective values to both programs are identical, then their respective solution vectors correspond to the unique solution of the actual QAP.*

Proof. The superiority of the respective SDP relaxation together with the incorporated *GLB* constraint over the plain *GLB* linearization is evident. For the remainder of the

proof, we assume identical objective values and uniqueness of the solution $\hat{X}$ to problem (6.1). By uniqueness of $\hat{X}$, we have

$$\forall X \in \mathcal{D}^n \backslash \Pi^n : \quad \langle L + C, X \rangle > \langle L + C, \hat{X} \rangle.$$

Moreover, (6.3) implies

$$\forall X \in \mathcal{D}^n : \quad \langle A, Y \rangle + \langle C, X \rangle \geqslant \langle L + C, X \rangle.$$

Taken together, these inequalities necessitate unequal objective values whenever the feasible point $(X, \ldots)$ to the given SDP relaxation does not correspond to an assignment. By assumption, it is therefore $X \in \Pi^n$. In this case, the validity of the conditional statement is an immediate consequence of Lemma 3.3 and Corollary 5.5. □

For nonzero approximation thresholds $\varepsilon > 0$, the respective level-1 relaxations do not inhere the characteristic stated in Corollary 5.5. We may restore this property by incorporating the following relaxation approach:

$$\begin{aligned} \inf_{M \in \mathcal{M}^n,\ X \in \Pi^n} \quad & \langle M, E \rangle + \langle C, X \rangle && \text{(6.5a)} \\ \text{s. t.} \quad & M \geqslant L \circ X, && \text{(6.5b)} \\ & M \geqslant U \circ X - U + AXB, && \text{(6.5c)} \end{aligned}$$

where L is defined as in (6.1b) and

$$U := (u_{ij}) \qquad \text{with} \quad u_{ij} = \max_{X \in \Pi,\ X_{ij} = 0} (AXB)_{ij}. \tag{6.5d}$$

The above linearization for the QAP was introduced by Xia and Yuan [109, 111].[1] They extended the Gilmore-Lawler bounding procedure by a modified version of the Kaufman and Broeckx's linearization [57], and proved that it inheres the desired property stated in Corollary 5.5, see [111, Theorem 3.7] and [109, Theorem 3].

The linearization approach by Xia and Yuan can be incorporated into the respective SDP relaxations by implementing

$$\operatorname{diag}(AY) \geqslant Me \tag{6.6}$$

[1] The formula for (u_{ij}) given in [111, Eq. (29)] is incorrect. It was corrected by Xia in [109, Eq. (4)].

together with the constraints (6.5b) and (6.5c). The extended integration into relaxations based on vector lifting techniques is similarly straightforward. Additional to (6.4), one applies the vector inequality

$$\operatorname{diag}((B^T \otimes A)\Upsilon) \geqslant \operatorname{vec}(U \circ X - U + AXB). \tag{6.7}$$

Many different linearization techniques can be incorporated by a very similar procedure. A typical approach to obtain tighter relaxations is the application of QAP reformulations. The procedures proposed by Assad and Xu [3] as well as Carraresi and Mallucelli [20], among many other works such as [19, 31, 35, 55, 91], demonstrate the influence of these reformulations on the quality of the Gilmore-Lawler bound. On the basis of numerical tests, we observed that QAP reformulations which are suitable for the discussed SDP relaxations can be less practical for *GLB* based constraints. We deal with this circumstance by implementing the corresponding conditions in consideration of a different QAP reformulation.

Assume that the matrix $\acute{L}$ is constructed as in (6.1b), but for a specific reformulation instance $(\acute{A}, \acute{B}, \acute{C})$ whose parameter matrices satisfy

$$\operatorname{diag}(\acute{A}) \equiv 0 \qquad \text{and} \qquad \acute{B} = B + \acute{v}_B e^T.$$

The Gilmore-Lawler bound linearizes the diagonal elements of these matrices in the same way as the considered relaxation frameworks. Moreover, adding a sum-matrix with constant columns to B has no effect on the bounding quality, see [34]. Hence, the presuppositions on the reformulated data matrices $\acute{A}$ and $\acute{B}$ serve just the purpose of simplicity and do not restrict the utility of the *GLB* conditions. The reformulated version of condition (6.3) is then

$$\acute{A}Y + \acute{A}X\acute{v}_B e^T \geqslant_{diag} \acute{L}X^T. \tag{6.8}$$

From the proof of Lemma 4.3, the adaptation of (6.4) is even more apparent:

$$\operatorname{diag}((\acute{B}^T \otimes \acute{A})\Upsilon) \geqslant \operatorname{vec}(\acute{L} \circ X). \tag{6.9}$$

Regarding the *VL* framework, there is no actual reason to choose different QAP reformulations for the objective function and the *GLB* inequalities. The consideration of this

case is nevertheless serviceable, because other constraints do benefit from different QAP reformulations.

Very similar inequalities can be derived for the off-diagonal elements of the matrix product $AXBX^T$. These may, for instance, be constructed by utilizing a slightly modified version of parameter L:

$$\underline{L} := (\underline{l}_{ij}) \qquad \text{with} \quad \underline{l}_{ij} = \min_{X \in \Pi,\, X_{ij}=0} (AXB)_{ij}.$$

By definition, we then have $AXBX^T \geqslant_{\text{off}} \underline{L}X^T$, which may be used for additional cuts in the corresponding frameworks. Unfortunately, numerical tests have shown that the improvement of the resulting bounds is negligible, whereas the impact on the computational effort is strongly apparent. The picture for the corresponding extension to the vector lifting based frameworks is even worse. The introduction of $n^4 - n^2$ additional inequality constraints penalizes the computation times significantly and the bounding improvement seems to dissolve within the accuracy of the used SDP solver. With regard to the efficiency of the relaxation program, we therefore limit our concern on the presented diagonal element inequalities.

Another possibility to acquire more constraints out of the Gilmore-Lawler bound is to split the corresponding inequalities in the manner of the discussed matrix-splitting schemes. By [62, Theorem 3.2], however, it is clear that the deduction of additional *GLB* inequalities via matrix splitting is generally not recommendable.

6.1.2. Eigenvalue related cuts

The possibility to construct additional constraints based on the Gilmore-Lawler bound procedure suggests the use of another well-known bounding technique, that is the eigenvalue based approach by Finke, Burkard, and Rendl [34].

We follow the notation in [34] and denote by $\langle v, w \rangle_+$ and $\langle v, w \rangle_-$ ordered dot products of real vectors $v, w \in \mathbb{R}^n$:

$$\langle v, w \rangle_+ := \langle v^{\downarrow}, w^{\downarrow} \rangle = \max_{X \in \Pi^n} \langle v, Xw \rangle, \quad \langle v, w \rangle_- := \langle v^{\downarrow}, w^{\uparrow} \rangle = \min_{X \in \Pi^n} \langle v, Xw \rangle, \tag{6.10}$$

where $w^{\downarrow}$ and $w^{\uparrow}$ denote the vectors to w whose elements are rearranged in non-ascending and non-descending order, respectively. The eigenvalue bound (*EVB*) is based on the

fact that

$$\forall X \in \Pi^n\colon \quad \big\langle \lambda(A), \lambda(B) \big\rangle_- \leqslant \big\langle A, XBX^T \big\rangle \leqslant \big\langle \lambda(A), \lambda(B) \big\rangle_+, \tag{6.11}$$

see [34, Theorem 3].

For the following discussion about eigenvalue related cuts, assume the eigenvalues of $A = \sum_{i=1}^n \mu_i p_i p_i^T$ to be sorted in non-ascending order, and in non-descending order denote by $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$ the eigenvalues of B. In [27, Chapter 2.2.2], Ding and Wolkowicz proposed a smart implementation for incorporating *EVB* into their matrix lifting based relaxation framework. They strengthened their relaxation by applying the cuts

$$0 \leqslant \sum_{i=1}^{l} \langle p_i, Y p_i \rangle - \lambda_i \qquad \text{for} \quad l \in \{1, \ldots, n-1\}. \tag{6.12}$$

From the proof of [27, Lemma 2.1], it is clear that (6.12) describes a sensible integration of *EVB* based conditions.

The incorporation into the respective SDP relaxations is straightforward. However, this does not mean that the presented procedure is similarly reasonable for all regarded relaxations frameworks. To illustrate this circumstance, consider the following result.

Lemma 6.2. *Let the QAP instance (A, B, C) be given and assume that the approximation tolerance ε is zero. For any feasible point $(X, F_+, F_-, U_1, \ldots, U_k, Y)$ to problem* (5.11)*, the majorization relation*

$$\lambda(Y) \prec \lambda(B) \tag{6.13}$$

holds valid.

Proof. Let $\lambda_\star := [\lambda_{1\star}, \ldots, \lambda_{k\star}]^T$ denote the vector consisting of the distinct eigenvalues $\{\lambda_{i\star}\}$ of B, and let $\{w_1, \ldots, w_n\}$ be a set of orthonormal eigenvectors of Y, such that $Y w_i = \lambda_i(Y) w_i$ for $1 \leqslant i \leqslant n$. Furthermore, define the $n \times k$ matrix $\check{S} := (\check{s}_{ij})$ with elements $\check{s}_{ij} = \langle w_i, U_j w_i \rangle$. Then,

$$\forall i \in \{1, \ldots, n\}\colon \quad \langle w_i, Y w_i \rangle = \sum_{i=j}^{k} \lambda_{j\star} \langle w_i U_j w_i \rangle = \sum_{j=1}^{k} \lambda_{i\star} \check{s}_{ij}$$

reveals the identity $\lambda(Y) = \check{S} \lambda_\star$.

The equality constraints in (5.11d) and (5.11f) imply

$$\forall i \in \{1,\ldots,n\}\colon \quad \sum_{j=1}^{k} \check{s}_{ij} = \langle w_i, \sum_{i=1}^{k} U_j w_i \rangle = \langle w_i, I w_i \rangle = 1$$

and

$$\forall j \in \{1,\ldots,k\}\colon \quad \sum_{i=1}^{n} \check{s}_{ij} = \operatorname{tr}([w_1,\ldots,w_n]^T U_j [w_1,\ldots,w_n]) = \operatorname{tr}(U_j) = |\Phi_j|.$$

Moreover, due to the positive semidefiniteness of the variables $\{U_j\}$, it follows $\check{S} \geqslant 0$. We complete the argument with the simple observation that the j-th column vector of $\check{S}$ can be written as the sum of $|\Phi_j|$ vectors whose elements are nonnegative and sum up to 1. The latter statement is valid for each column of $\check{S}$ and implies the existence of a doubly stochastic matrix S that satisfies $\check{S}\lambda_\star = S\lambda(B)$. This, in turn, validates the identity $\lambda(Y) = S\lambda(B)$ for some $S \in \mathcal{D}^n$. □

For an arbitrary set of orthonormal basis vectors $\{w_1,\ldots,w_n\}$ spanning $\mathbb{R}^n$, define the orthogonal matrix $W := [w_1,\ldots,w_n]$. By Theorem 2.6 and Lemma 6.2, we then derive the majorization relation

$$\operatorname{diag}(W^T Y W) \prec \lambda(W^T Y W) = \lambda(Y) \prec \lambda(B).$$

Thus, the observation that the eigenvalues of any feasible matrix variable Y to problem (5.11) are majorized by the eigenvalues of B implies the compliance of Y with the inequalities

$$\sum_{i=1}^{l} \langle w_i, Y w_i \rangle \geqslant \operatorname{tr}(Y) - \sum_{j=l+1}^{n} \operatorname{diag}_j^{\downarrow}(W^T Y W) \geqslant \operatorname{tr}(Y) - \sum_{j=l+1}^{n} \lambda_j^{\downarrow}(B) = \sum_{i=1}^{l} \lambda_i,$$

where $\operatorname{diag}_j^{\downarrow}(\cdot)$ denotes the j-th largest diagonal element of the corresponding matrix. Since this relation holds valid for arbitrary choices of orthonormal bases spanning $\mathbb{R}^n$, this naturally includes the set of eigenvectors of A. In this respect, the integration of *EVB* based constraints such as (6.12) into *ESC* is redundant. By the arguments for Theorem 3.2 and Corollary 5.4, we further derive the same conclusion for *ES* and *VL*.

Even for the SDP relaxation with the smallest dimension, QAP_{MS}, it is sufficient to incorporate only a subset of the inequalities in (6.12). The distribution of the positive and negative eigenvalues of B provides the opportunity to construct a stronger and

more efficient version of (6.12). For relaxation frameworks that utilize the PSD splitting defined in (3.9), we show the following result.

Lemma 6.3. *For the parameter matrix B of a given QAP instance (A, B, C), let (B_+, B_-) denote the PSD splitting defined in* (3.9)*. Additionally, let r_+ and r_- denote the ranks of the matrices B_+ and B_-, respectively. If incorporated into the corresponding instance of relaxation* (4.32)*, then*

$$-\sum_{i=1}^{l}\langle p_i, Y_- p_i\rangle \geqslant \sum_{i=1}^{l}\lambda_i, \qquad 1 \leqslant l < r_-, \tag{6.14a}$$

together with

$$\sum_{i=1}^{l}\langle p_i, Y_+ p_i\rangle \geqslant \sum_{i=n-r_+}^{l}\lambda_i, \qquad n - r_+ < l < n, \tag{6.14b}$$

imply the validity of all inequalities in (6.12).

Proof. Regarding the first $r_- - 1$ inequalities, $1 \leqslant l < r_-$, the positive semidefiniteness of $Y_+ = Y + Y_-$ and (6.14a) require

$$\sum_{i=1}^{l}\langle p_i, Y p_i\rangle \geqslant -\sum_{i=1}^{l}\langle p_i, Y_- p_i\rangle \geqslant \sum_{i=1}^{l}\lambda_i.$$

Furthermore, the orthogonality of the eigenvectors $\{p_i\}$ implies

$$\forall l \in \{1, \ldots, n\}\colon \quad \sum_{i=1}^{l}\langle p_i, Y p_i\rangle \geqslant -\langle p_i, Y_- p_i\rangle \geqslant -\operatorname{tr}(Y_-) = -\operatorname{tr}(B_-). \tag{6.15}$$

By definition, we have $\lambda_1 \leqslant \ldots \leqslant \lambda_{r_-} < 0 = \lambda_{r_-+1} = \ldots = \lambda_{n-r_+} < \lambda_{n-r_++1} \leqslant \ldots \leqslant \lambda_n$. Therefore (6.15) validates

$$\sum_{i=1}^{l}\langle p_i, Y p_i\rangle \geqslant -\operatorname{tr}(B_-) = \sum_{i=1}^{l}\lambda_i \qquad \text{for} \quad r_- \leqslant l \leqslant n - r_+.$$

Finally, adding (6.15) and (6.14b) yields

$$\sum_{i=1}^{l}\langle p_i, Y p_i\rangle = \sum_{i=1}^{l}\langle p_i, Y_+ p_i\rangle - \sum_{i=1}^{l}\langle p_i, Y_- p_i\rangle \geqslant \sum_{i=n-r_+}^{l}\lambda_i - \operatorname{tr}(B_-) = \sum_{i=1}^{l}\lambda_i$$

for $n - r_+ < l < n$, which finishes the argument. □

By using (6.14a) and (6.14b), we realize a tighter version of the discussed bounding technique necessitating only $\operatorname{rank}(B) - 2$ inequality constraints instead of the original $n - 1$ conditions. At a first glance, the reduction of the framework by not more than $n - \operatorname{rank}(B) + 1$ linear inequality constraints may be hardly worth the effort of elaborating the specific implementation details. Nevertheless, the influence on the solving procedure should not be underestimated. Each of these inequalities introduces n^2 or $\binom{n+1}{2}$ coefficients to the actual SDP data, respectively. In regard to the memory management of the applied solver, the number of coefficients can be quite important for the performance of the solving procedure.

For the actual implementation of the discussed *EVB* cuts, there are more details that deserve our attention. As already described for *GLB* based constraints, also *EVB* based ones like (6.12) can be modified for different reformulations of the actual quadratic assignment problem. Reduction rules to derive appropriate reformulations have been elaborated, for example, in [34, 44, 88]. In the final version of their matrix lifting based SDP relaxation [27, MSDR_3], Ding and Wolkowicz applied their *EVB* based constraints to a projected reformulation of the QAP. By [27, Lemma 2.2], it was moreover shown that the corresponding relaxation incorporates the projection bound (*PB*) introduced in [44].

Hadley, Rendl, and Wolkowicz demonstrated in [44] that *PB* outperforms *EVB1* for all tested QAP instances. In consideration of the interaction between the actual eigenvalue bound and the respective SDP relaxation in which this bound shall be incorporated, numerical tests for a wider range of problems taken from the QAP library [18] showed a slightly different picture. As a suitable integration in the respective SDP frameworks the author suggests the straightforward utilization of the reformulated QAP instance defined in (4.20). Actually, maybe not completely straightforward. The effect of the inequality conditions in (6.12) can be improved by a slight modification to our initial presuppositions on the eigenvalues and eigenvectors of A and B. For this purpose, we exploit our knowledge about the presence of the particular eigenvector $\frac{1}{\sqrt{n}}e$. Since this vector is unaffected by permutations and the corresponding eigenvalue is equal to zero, it is possible to remove it from the *EVB* based inequalities. Let the index to this specific eigenvalue-eigenvector pair be fixed to $i = 1$, and let all other eigenvalue-eigenvector pairs satisfy the general presuppositions for this Subsection. In this context, A and B

may be written as

$$A = \sum_{i=2}^{n} \mu_i p_i p_i^T, \qquad \mu_2 \geqslant \mu_3 \geqslant \ldots \geqslant \mu_n, \qquad \langle e, p_i \rangle = 0|_{2 \leqslant i \leqslant n} \tag{6.16a}$$

and

$$B = \sum_{i=2}^{n} \lambda_i q_i q_i^T, \qquad \lambda_2 \leqslant \lambda_3 \leqslant \ldots \leqslant \lambda_n, \qquad \langle e, q_i \rangle = 0|_{2 \leqslant i \leqslant n}, \tag{6.16b}$$

where $\mu_1 = \lambda_1 = 0$ and $p_1 = q_1 = \frac{1}{\sqrt{n}} e$. If we apply these adjusted index assignments, then

$$0 \leqslant \sum_{i=2}^{l} p_i^T Y p_i - \lambda_i \qquad \text{for} \quad l \in \{2, \ldots, n-1\}, \tag{6.17}$$

states a tighter and more economic version of (6.12).

For constraints of the form

$$0 \leqslant \sum_{i=1}^{l} w_i^T Y w_i - \lambda_i \qquad \text{for} \quad l \in \{1, \ldots, n-1\},$$

it is evident that the choice of the basis vectors $\{w_1, \ldots, w_n\}$ has a significant influence on the bounding quality. Considering the objective function $\langle A, Y \rangle + \langle C, X \rangle$, the choice of the eigenvectors of A is reasonable since it incorporates the corresponding eigenvalue bound. Nevertheless, this choice may not necessarily be the best possible one. A very similar argument as the one we used to explain the choice of the reformulation vector d_B given in (4.20) is also applicable to a reformulation of the matrix A.

The reformulation vectors d_A and v_A defined in (4.20) are designed to minimize the Frobenius norm of the reformulated data matrix A. For a strong eigenvalue bound this approach is reasonable but can be improved. The last statement is evident from superior performance of the bounding techniques *PB* [44] and *EVB2* [88] compared to *EVB1* [88]. Instead of simply taking over one of these approaches, we exploit the idea of weighted positive and negative semidefinite parts of A. More specifically, we utilize a splitting approach for A which is weighted in regard to the eigenvalue distribution of B. The corresponding adaptation of problem (4.18) is given by

$$\begin{aligned} \inf_{d_A, v_A \in \mathbb{R}^n,\ A_1, A_2 \in \mathcal{S}_+^n} \quad & \left\| \alpha_1 A_1 + \alpha_2 A_2 \right\|_{\mathrm{F}} \\ \text{s. t.} \quad & A + \operatorname{diag}^*(d_A) + v_A e^T + e v_A^T = A_1 - A_2, \end{aligned} \tag{6.18}$$

where the weighting coefficients α_1 and α_2 are defined in respect of the eigenvalues of B:

$$\alpha_1 := \sqrt{\sum_{i:\,\lambda_i<0} \lambda_i^2} \qquad \text{and} \qquad \alpha_2 := \sqrt{\sum_{i:\,\lambda_i>0} \lambda_i^2}.$$

By solving problem (6.18), we obtain new reformulation vectors d_A and v_A. Since the eigenspace of the corresponding reformulation of A is often more advantageous to compute tight eigenvalue bounds, we utilize the eigenvalue decomposition of

$$\acute{A} = A + \overset{*}{\operatorname{diag}}(d_A) + v_A e^T + e v_A^T = \sum_{i=1}^{n} \acute{\mu}_i \acute{p}_i \acute{p}_i^T \tag{6.19}$$

to obtain the basis vectors $(w_1, \ldots, w_n) = (\acute{p}_1, \ldots, \acute{p}_n)$. For these vectors, we assume that the ordering of the eigenvalues $\{\acute{\mu}_i\}$ satisfies our presuppositions in (6.16).

If the respective SDP relaxation is used within a branch-&-bound algorithm, it is possible to attain more beneficial sets of basis vectors $\{w_1, \ldots, w_n\}$ in a significantly more efficient way. The approach is as follows: suppose that the respective SDP relaxation has already been computed for different subproblems of the considered QAP. From the pool of already solved SDP relaxations, choose the instance which is most similar to the problem that needs to be solved in the current bounding step. Instead of the eigenvectors of the (possibly reformulated) coefficient matrix A, utilize the eigenvalue decomposition of the matrix $\hat{Y}$ obtained from the solution vector to the chosen problem instance. Order the eigenvectors with respect to the accompanied eigenvalues of $\hat{Y}$ and apply the necessary adaptations for the applicability to the current relaxation instance. The latter step may involve the transformation into another space.

By allowing higher efforts on the implementation as well as the computations, it is possible to strengthen the *EVB* based cuts. In that context, let us consider the convex quadratic programming framework *SOCPB* introduced in [110]. For the construction of this relaxation, Xia uses the identity

$$\begin{aligned} \operatorname{tr}(AXBX^T) &= \operatorname{tr}\left(\sum_{i=1}^{n} \mu_i p_i p_i^T \Big(\sum_{j=1}^{n} \lambda_j X q_i q_i X^T \Big) \right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \mu_i \lambda_j \operatorname{tr}(p_i p_i^T X q_i q_i X^T) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \mu_i \lambda_j \langle p_i, X q_j \rangle^2. \end{aligned} \tag{6.20}$$

He defines a matrix $S := (s_{ij})$ with $s_{ij} = \langle p_i, Xq_j\rangle^2$ for $1 \leqslant i, j \leqslant n$, and describes a relaxation of the corresponding quadratic equalities via

$$\mathsf{s}_{ij} \geqslant \langle p_i, X q_j\rangle^2, \qquad 1 \leqslant i, j \leqslant n, \tag{6.21}$$

together with the equality constraints that realize $\mathsf{S} \in \mathcal{E}^n$. The latter condition is an immediate consequence of the orthogonality of $\{p_i\}$ and $\{q_j\}$, yielding

$$\forall X \in \Pi^n, j \in \{1, \ldots, n\}: \quad \sum_{i=1}^{n} \langle p_i, Xq_j\rangle^2 = \|[p_1, \ldots, p_n]^T X q_j\|^2 = \|q_j\|^2 = 1$$

and

$$\forall X \in \Pi^n, i \in \{1, \ldots, n\}: \quad \sum_{j=1}^{n} \langle p_i, Xq_j\rangle^2 = \|p_i^T X [q_1, \ldots, q_n]\|^2 = \|p_i\|^2 = 1.$$

For the integration into the respective SDP relaxation, we introduce the same matrix variable S, add the corresponding equality constraints for $\mathsf{S} \in \mathcal{E}^n$ together with the inequalities in (6.21), and exploit the identities

$$p_i^T X B X^T p_i = p_i^T X \left(\sum_{j=1}^{n} \lambda_j q_j q_j^T \right) X^T p_i = \sum_{j=1}^{n} \lambda_j \langle p_i, Xq_j\rangle^2, \qquad 1 \leqslant i, j \leqslant n,$$

to link the variables Y and S via the following equality conditions

$$p_i^T \mathsf{Y} p_i = \sum_{j=1}^{n} \lambda_j \mathsf{s}_{ij}, \qquad 1 \leqslant i, j \leqslant n. \tag{6.22}$$

From the proof of Lemma 6.2, it is clear that the incorporation of these conditions into *ESC*, *ES*, or *VL* is redundant, at least if we assume $\varepsilon = 0$. Additional upper bound constraints on the variables $\{\mathsf{s}_{ij}\}$ can change this. In order to attain a further tightening of the framework *SOCPB*, Xia utilizes the following linear upper bounds

$$\forall i, j \in \{1, \ldots, n\}: \quad (l_{ij} + u_{ij}) p_i^T \mathsf{X} q_j - l_{ij} u_{ij} \geqslant \mathsf{s}_{ij}, \tag{6.23}$$

where $l_{ij} := \langle p_i, q_j\rangle_-$ and $u_{ij} := \langle p_i, q_j\rangle_+$ define lower and upper bounds of the corresponding linear terms $\{p_i^T \mathsf{X} q_j\}$, respectively.

We derive similar upper bounds as in (6.23) by exploiting the following identities

$$\begin{aligned}\forall X \in \Pi^n,\, i,j \in \{1,\dots,n\}\colon \quad p_i^T X q_j &= \sum_{k=1}^{n} p_i^T e_k e_k^T X q_j \\ &= \sum_{k=1}^{n} -|p_i^T e_k e_k^T X q_j| + 2\max\{p_i^T e_k e_k^T X q_j, 0\} \\ &= -|p_i|^T X |q_i| + 2\sum_{k=1}^{n} \max\{p_i^T e_k e_k^T X q_j, 0\},\end{aligned}$$

and

$$\forall X \in \Pi^n,\, i,j \in \{1,\dots,n\}\colon \quad p_i^T X q_j = |p_i|^T X |q_i| + 2\sum_{k=1}^{n} \min\{p_i^T e_k e_k^T X q_j, 0\}.$$

Together with the limits of the respective sum terms

$$\begin{aligned}\delta_{ij}^l &:= \min_{X\in\Pi^n} \left\{ 2\sum_{k=1}^{n} \max\{p_i^T e_k e_k^T X q_j, 0\} \right\}, \\ \delta_{ij}^u &:= \max_{X\in\Pi^n} \left\{ 2\sum_{k=1}^{n} \min\{p_i^T e_k e_k^T X q_j, 0\} \right\},\end{aligned} \tag{6.24}$$

we obtain new linear bounding constraints:

$$\max\{u_{ij}, -l_{ij}\}\, |p_i|^T X |q_i| + \max\{u_{ij}\delta_{ij}^u, l_{ij}\delta_{ij}^l\} \geqslant s_{ij}, \qquad 1 \leqslant i,j, \leqslant n. \tag{6.25}$$

In this context, it is worth mentioning that the necessary computations for the values defined in (6.24) can be realized very efficiently via

$$\delta_{ij}^l = \langle p_i^{\downarrow}, q_j^{\uparrow}\rangle + \langle |p_i^{\downarrow}|, |q_j^{\uparrow}|\rangle \qquad \text{and} \qquad \delta_{ij}^u = \langle p_i^{\downarrow}, q_j^{\downarrow}\rangle - \langle |p_i^{\downarrow}|, |q_j^{\downarrow}|\rangle.$$

Moreover, by introducing intermediate variables for the terms $\{X|q_j|\}$ (alternatively $\{|p_i|^T X\}$), it is possible to reduce the number of nonzero coefficients that are necessary for the implementation of (6.25) to about $2n^3$.

Although (6.25) does not imply the validity of (6.23) - meaning that (6.25) is not strictly tighter than (6.23) - the former performs in general significantly better. This statement is particularly true if the respective constraints are incorporated into one of the discussed SDP frameworks.

If the computational costs are of minor importance, it is possible to use the even stronger upper bounds:

$$\langle \bar{M}_{ij}, X\rangle + \max\{\bar{\delta}^u_{ij}, \bar{\delta}^l_{ij}\} \geqslant s_{ij}, \qquad 1 \leqslant i, j \leqslant n, \tag{6.26}$$

where $\bar{M}_{ij} := \max(l_{ij} p_i q_j^T, u_{ij} p_i q_j^T)$ are defined as the element-wise maxima of the corresponding rank-1 parameter matrices, and

$$\bar{\delta}^l_{ij} := \max_{X \in \Pi^n} \langle l_{ij} p_i q_j^T - \bar{M}_{ij}, X\rangle, \qquad \bar{\delta}^u_{ij} := \max_{X \in \Pi^n} \langle u_{ij} p_i q_j^T - \bar{M}_{ij}, X\rangle$$

define the corresponding adaptations to the offset corrections in (6.24). The respective coefficient matrices still have low ranks, providing similar opportunities for the reduction of the computational costs like the ones we indicated for the implementation of (6.25). Nevertheless, due to the absence of reiterations in the corresponding computations, the author has not been able to reduce the computational complexity below $\mathcal{O}(n^3 \log n)$. In respect of the small influence on the tightness of the considered SDP relaxations and the significantly greater computational effort, the constraints in (6.25) seem preferable to the ones in (6.26).

If we are concerned with larger QAP instances, even the constraints in (6.23) and (6.25) seem rather impractical. Though it is possible to realize a deep integration into *ES* and *VL*, the additional effort does not pay off in the same way for other relaxations frameworks. By combining the approach in (6.17) with some of the bounds in (6.25), it is possible to obtain a very efficient integration of the eigenvalue bound. Let $\{\bar{s}_{ij}\}$ denote upper bounds for the respective quadratic terms $\{\langle p_i, X q_j\rangle^2\}$. For the eigenvector p_2 of A, it is easy to see that

$$p_2^T X B X^T p_2 \geqslant \lambda_2 \langle p_2, X q_2\rangle^2 + \lambda_3 \left(1 - \langle p_2, X q_2\rangle^2\right) \geqslant \lambda_2 \bar{s}_{22} + \lambda_3 (1 - \bar{s}_{22}).$$

With $\bar{s}_{2:3} := \min\{\bar{s}_{22}, \bar{s}_{33}\}$, we educe the inequality for the sum over the first two terms:

$$\sum_{i=2}^{3} p_i^T X B X^T p_i \geqslant \lambda_2 \bar{s}_{2:3} + \lambda_3 (1 - \bar{s}_{2:3}) + \lambda_2 (1 - \bar{s}_{2:3}) + \lambda_3 \bar{s}_{2:3} = \lambda_2 + \lambda_3.$$

This matches the second inequality in (6.17). The third condition may then again be improved:

$$\sum_{i=2}^{4} p_i^T X B X^T p_i \geqslant \lambda_2 + \lambda_3 + \lambda_4 \bar{s}_{44} + \lambda_5 (1 - \bar{s}_{44}).$$

It is therefore recommendable to replace every second inequality in (6.17) by

$$(\lambda_{l+1} - \lambda_l)(1 - \bar{s}_{ll}) \leqslant \sum_{i=2}^{l} p_i^T Y p_i - \lambda_i \qquad \text{for} \quad l = 2, 4, 6, \ldots . \tag{6.27}$$

Appropriate terms for $\bar{s}_{ll}$ can be taken from (6.23), (6.25), or even (6.26). For a minimal computational costs, one may simply use $\bar{s}_{ll} = \max\{u_{ll}^2, l_{ll}^2\}$.

6.1.3. Linear bound constraints

For many relaxations instances, it is possible to attain a significant improvement of the bounding quality by applying additional bounds to its optimization variables. In [73] and [83], Mittelmann, Peng, and Li introduced new inequality constraints based on symmetric functions [70].

Definition 6.4. *A function* $\mathrm{f}(v)\colon \mathbb{R}^n \to \mathbb{R}$ *is said to be symmetric if for any permutation matrix* $X \in \Pi^n$*, the relation* $\mathrm{f}(v) = \mathrm{f}(Xv)$ *holds.*

One of these functions, namely the additive function $\mathrm{f}(v) = \langle e, v \rangle$, has already been used for the constraints (3.3e), (3.5d), (3.10d), and (3.11d). Other symmetric functions, that are useful for the construction of valid constraints, are the minimum and the maximum function as well as p-norms:

$$\forall v \in \mathbb{R}^n\colon \quad \min(v) = \min_{1 \leqslant i \leqslant n} v_i, \quad \max(v) = \max_{1 \leqslant i \leqslant n} v_i, \quad \|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{\frac{1}{p}}.$$

If these operators are applied to a matrix $M \in \mathcal{M}^{m,n}$, they act along the rows of the respective matrix, i.e.

$$\min(M) = \big(\min(e_1^T M),\ \min(e_2^T M),\ \ldots,\ \min(e_m^T M)\big)^T.$$

In [73], [83], [84], and also [25], the minimum and maximum functions are used to obtain linear bounds for several optimization variables and linear combinations of these. The

corresponding constraints on the matrix variable Y have the form

$$(X \min(B))_i \leqslant (Y)_{ij} \leqslant (X \max(B))_i, \qquad 1 \leqslant i, j \leqslant n. \tag{6.28}$$

For a further tightening of the respective relaxations, Peng, Mittelmann, and Li [83] applied the same kind of constraints to each matrix variable Y_+ and Y_- as well as their sum. When it comes to the relaxations *ES* and *ESC*, it is possible to exploit this approach to the extreme by using all of their matrix variables and various linear combinations of these. In this subsection, however, we are not so much interested in applying these inequalities to different linear combinations of the respective matrix variables. We are mainly concerned with investigating possible improvements of the corresponding constraints.

Denote the vectors consisting of the minimal and maximal row elements of B by $v_{\min} := \min(B)$ and $v_{\max} := \max(B)$, respectively. Condition (6.28) may also be stated in the following form:

$$X v_{\min} e^T \leqslant Y \leqslant X v_{\max} e^T.$$

By the nonnegativity of X, clearly $v_{\min} e^T \leqslant B \leqslant v_{\max} e^T$ implies

$$X v_{\min} e^T = X v_{\min} e^T X^T \leqslant X B X^T \leqslant X v_{\max} e^T X^T = X v_{\max} e^T,$$

and thus yields (6.28). The last observation motivates a further exploitation of sum-matrix inequalities to obtain tighter constraints. Define, for instance,

$$w_{\min} := \min(B^T - e v_{\min}^T) \qquad \text{and} \qquad w_{\max} := \max(B^T - e v_{\max}^T).$$

By definition, we have $v_{\min} e^T + e w_{\min}^T \leqslant B \leqslant v_{\max} e^T + e w_{\max}^T$, which leads to the inequality constraints

$$X v_{\min} e^T + e w_{\min}^T X^T \leqslant Y \leqslant X v_{\max} e^T + e w_{\max}^T X^T.$$

Since $w_{\min} \geqslant 0$ and $w_{\max} \leqslant 0$, it is apparent that these bounds are at least as good as the ones in (6.28).

For the linear inequalities based on the minimum or the maximum function, Mittelmann and Peng [73] pointed out that - since the diagonal elements of Y_+ and Y_- are already described by the corresponding equality constraints - it is sufficient to consider

solely the off-diagonal variables. We further observe that, due to the symmetry of B, the symmetric parts of the respective sum-matrices satisfy the same bounding conditions, i.e.

$$ve^T + ew^T \leqslant_{\text{off}} B \quad \Longrightarrow \quad \tfrac{1}{2}(v+w)e^T + \tfrac{1}{2}e(v+w)^T \leqslant_{\text{off}} B. \tag{6.29}$$

Let the gap between a sum-matrix $ve^T + ew^T$ and an arbitrary real matrix $B = (b_{ij})$ of the same dimension be defined as

$$\delta_{\text{gap}}(B, v, w) := \sum_{\substack{i,j \\ i \neq j}} |b_{ij} - v_i - w_j| = \langle E_{\text{off}}, |B - ve^T - ew^T| \rangle. \tag{6.30}$$

A suitable approach to obtain tight sum-matrix inequalities is the minimization of the respective gaps. By $\delta_{\text{gap}}(B, v, w) = \delta_{\text{gap}}(B, \frac{1}{2}(v+w), \frac{1}{2}(v+w))$ and the implication in (6.29), it is apparently sufficient to concentrate on the strictly lower triangular elements of symmetric sum-matrices. The following linear programming problem can be used to compute lower and upper symmetric sum-matrix bounds for B that accompany minimal gaps:

$$\begin{aligned} \inf_{\mathsf{v}_l, \mathsf{v}_u \in \mathbb{R}^n} \quad & \langle e, \mathsf{v}_u - \mathsf{v}_l \rangle \\ \text{s. t.} \quad & \mathsf{v}_l e^T + e\mathsf{v}_l^T \leqslant_{\text{tri}} B \leqslant_{\text{tri}} \mathsf{v}_u e^T + e\mathsf{v}_u^T. \end{aligned} \tag{6.31}$$

Symmetric sum-matrix bounds have a big advantage over their non-symmetric equivalents. Due to the symmetry, they require only half as many LP inequalities. Indeed, quite often there exists no sum-matrix bound that is not symmetric and involves the same optimal gap as the solution to problem (6.31). On the other hand, sum-matrix bounds with the same symmetric part but noticeable skew-symmetric components yield tighter inclusions.

A significant skew symmetric part requires the computation of dissimilar parameter vectors v and w. Unfortunately, the maximization of some p-norm difference between these vectors leads to a concave optimization problem. For this reason, it seems advantageous to switch to other optimization criteria. Here, we utilize the following program

$$\begin{aligned} \inf_{\mathsf{v}_l, \mathsf{w}_l, \mathsf{v}_u, \mathsf{w}_u \in \mathbb{R}^n} \quad & \langle \hat{v}_l, \mathsf{v}_l \rangle + \langle \hat{v}_u, \mathsf{v}_u \rangle \\ \text{s. t.} \quad & \mathsf{v}_l e^T + e\mathsf{w}_l^T \leqslant_{\text{off}} B \leqslant_{\text{off}} \mathsf{v}_u e^T + e\mathsf{w}_u^T, \\ & \langle e, \mathsf{v}_l \rangle = \langle e, \mathsf{w}_l \rangle = \langle e, \hat{v}_l \rangle, \quad \langle e, \mathsf{v}_u \rangle = \langle e, \mathsf{w}_u \rangle = \langle e, \hat{v}_u \rangle, \end{aligned} \tag{6.32}$$

where the vector coefficients $\hat{v}_l$ and $\hat{v}_u$ are obtained by solving problem (6.31). The computed parameter vectors v_l, w_l, v_u, and w_u can then be used to construct non-symmetric sum-matrix inequalities of the form

$$X v_l e^T + e w_l^T X^T \leqslant_{\text{off}} Y \leqslant_{\text{off}} X v_u e^T + e w_u^T X^T. \tag{6.33}$$

Obviously, there is nothing to gain by applying adapted sum-matrix bounds to reformulated versions of the same problem instance. Suitable approaches for a further tightening of these bounds are the application of multiple varying sum-matrix inequalities or the construction of the same type of bounds for linear combinations of the respective matrix variables. In consideration of the "eigenspace" SDP relaxation, for example, it is possible to create linear bounds for each matrix variable Q_i. The number of applicable bounds is virtually endless if we consider arbitrary linear combinations of these.

In a very similar way, one can derive linear bounds for the lifted variable Y in problem (3.3). For some $X \in \Pi^n$ and the corresponding rank-1 matrix $\Upsilon = \operatorname{vec}(X)\operatorname{vec}(X)^T$, we have

$$\Upsilon = (I \otimes X)^T \operatorname{vec}(I)\operatorname{vec}(I)^T (I \otimes X) \qquad \text{with} \quad I \otimes X \in \Pi^{n^2}.$$

The lower sum-matrix bound for $\operatorname{vec}(I)\operatorname{vec}(I)^T$ that accompanies the smallest possible gap is obtained for $v_l = w_l = 0_{(n^2)}$. By utilizing these vectors for the respective sum-matrix bound, we derive the inequality condition

$$Y \geqslant (I \otimes X)^T v_l e_{(n^2)}^T + e_{(n^2)} w_l^T (I \otimes X) \equiv 0. \tag{6.34}$$

The same approach may be used to construct upper bounds on the variable Y. For those, however, it can be shown that they are redundant.

Though the way how (6.34) was established is rather uncommon, we used this explication because it is consistent to the previous explanations. A more natural deduction of the element-wise inequality $Y \geqslant 0$ is the inheritance of this property from its factors: $X \in \mathcal{N}^n \implies \Upsilon \in \mathcal{N}^{n^2}$. As a consequence of this natural deduction, the particular relaxation design VL_0 supplemented by the constraint $Y \geqslant 0$ has been investigated in many different research papers. In [116], for instance, the respective relaxation is referred to as QAP_{R3}.

Due to nearly $\frac{1}{2}n^4$ non-redundant inequality conditions, the incorporation of (6.34) is very expensive. On the other hand, in comparison to its low-dimensional counterparts, (6.34) is clearly superior. In order to show this, let us recall the connection between the variable Υ and corresponding subsets of feasible points to the other presented relaxation frameworks. In the proof of Theorem 3.2, we generated feasible instances for the variables $\{Q_i\}$ used in problem (3.11) out of a matrix Υ that satisfies the constraints of problem (3.3). This was done by using the identities

$$\check{Q}_i = (q_i \otimes I)^T \Upsilon (q_i \otimes I) = (e \otimes I)^T \left(\Upsilon \circ (q_i q_i^T \otimes E)\right)(e \otimes I), \qquad 1 \leqslant i \leqslant n.$$

In Subsection 4.3, we proceeded similarly to introduce the variable Y into relaxation VL_1, see (4.30) and the corresponding equality constraints in (4.29e). By the proofs of Theorem 3.2 and Corollary 5.4, it is clear that any feasible Y in problem (4.29) can be used to generate feasible variables to the other level-1 relaxations. The same procedure can be applied in order to construct feasible points for all SDP frameworks that have been considered until this point.

Consider, for instance, a lower sum-matrix bound to variable $Y_\triangle$ used in *IIMS*:

$$X v_\triangle e^T + e w_\triangle^T X^T \leqslant Y_\triangle.$$

The relation $v_\triangle e^T + e w_\triangle^T \leqslant B_\triangle$ and the nonnegativity of Υ implies

$$\begin{aligned} X v_\triangle e^T + e w_\triangle^T X^T &= (e \otimes I)^T \left(\Upsilon \circ \left((v_\triangle e^T + e w_\triangle^T) \otimes E\right)\right)(e \otimes I) \\ &\leqslant (e \otimes I)^T \left(\Upsilon \circ (B_\triangle \otimes E)\right)(e \otimes I) =: \check{Y}_\triangle, \end{aligned}$$

where $\check{Y}_\triangle$ denotes the generated instance for the variable $Y_\triangle$, thereby satisfies all constraints of problem (5.29). By the same argument, we conclude the compliance with all other sum-matrix bounds.

6.1.4. Cuts based on *p*-norm conditions

The third type of constraints that can be derived from symmetric functions are norm conditions on the rows and columns of the respective matrix variables. In [83], [84], and [25], the introduced SDP relaxations are tightened via additional norm constraints

of the form

$$\mathcal{L}_2(Y) \leqslant X\,\mathcal{L}_2(B), \tag{6.35}$$

where $\mathcal{L}_p(B)$ denotes the column vector whose components are the p-norms of the corresponding rows of B,[2] i.e.

$$\mathcal{L}_p(B) = \left(\|e_1^T B\|_p,\ \|e_2^T B\|_p,\ \ldots,\ \|e_m^T B\|_p\right)^T. \tag{6.36}$$

In our attempt to enhance these norm conditions, we made three discoveries. Firstly, the semidefiniteness condition in (3.5b) necessitates the validity of the vector inequality $\operatorname{diag}(Z) \geqslant \operatorname{diag}(Y^2)$, such that

$$\mathcal{L}_2(Y) \circ \mathcal{L}_2(Y) = \operatorname{diag}(Y^2) \leqslant \operatorname{diag}(Z) = X \operatorname{diag}(B^2) = X\left(\mathcal{L}_2(B) \circ \mathcal{L}_2(B)\right). \tag{6.37}$$

The inequality in (6.37) implies $\|\!|Y|\!\|_{\mathrm{F}} \leqslant \|\!|B|\!\|_{\mathrm{F}}$ and may be interpreted as a squared version of (6.35). Though this does not imply the redundancy of the constraint in (6.35), it is a good indication for a moderate effect on the feasible set. This connection is mirrored in different numerical examples. Naturally, this statement is also true for all SDP relaxation which are provably tighter than *ML*. This includes all versions of *MLX*, *ESC*, *ES*, and *VL*.

Our second discovery concerns the effect of the applied reformulation approach. A suitable idea to tighten the respective norm cuts is based on the construction of a reformulated matrix $B + v_p e^T + e w_p^T$ which accompanies minimal p-norm values for each of its rows. This may be realized via

$$\inf_{v_p, w_p \in \mathbb{R}^n} \quad \left\langle e, \mathcal{L}_p((B + v_p e^T + e w_p^T)_{\mathrm{off}}) \right\rangle.$$

The reformulated version of the respective norm constraints is then given by

$$\mathcal{L}_p((Y + X v_p e^T + e w_p^T X^T)_{\mathrm{off}}) \leqslant X\,\mathcal{L}_p((B + v_p e^T + e w_p^T)_{\mathrm{off}}). \tag{6.38}$$

In the absence of other constraints, the application of this procedure usually improves the bounding quality of the corresponding relaxation. However, incorporated into the full relaxation frameworks together with the other applicable constraints, the influence is

[2] This notion is taken over from the cited papers. It should not be confused with Lebesgue spaces.

marginal and the effect often reversed. For this reason, we advice against using additional QAP reformulations.

The final observation is based on numerical tests for different p-norm conditions. Amongst the tested p-norm conditions, the strongest bounds were usually obtained for $p = 1$. Compared to many other p-norm conditions, it is furthermore possible to implement 1-norm inequalities quite efficiently. They are therefore well suited for the incorporation into the respective SDP relaxation. On the other hand, if the applied solver has direct support for second order cone programming constraints, 2-norm conditions are handled even more efficiently.

In order to achieve tight relaxations, we combine both types of norm constraints. However, instead of a straightforward implementation of

$$\mathcal{L}_1(Y) \leqslant X\,\mathcal{L}_1(B) \qquad \text{and} \qquad \mathcal{L}_2(Y) \leqslant X\,\mathcal{L}_2(B), \tag{6.39}$$

we design constraints that accompany a stronger interrelation between these constraints.

Let B_{plus} define the matrix that is obtained from B by setting its diagonal elements as well as all negative entries to zero, i.e. $B_{\text{plus}} := (\max\{b_{ij}, 0\})_{\text{off}}$. In the actual implementation, we introduce a symmetric matrix $Y_{\text{plus}} \in \mathcal{S}^n \cap \mathcal{N}^n$ with only nonnegative entries. Additionally, we assume that the diagonal elements of Y_{plus} are equal to zero, which means that Y_{plus} is characterized by $\binom{n}{2}$ nonnegative variables. This matrix is used to relax the quadratic term $XB_{\text{plus}}X^T$, thereby complies with the conditions

$$Y_{\text{plus}} \geqslant_{\text{tri}} Y \qquad \text{and} \qquad Y_{\text{plus}}e = XB_{\text{plus}}e. \tag{6.40}$$

Instead of applying the 2-norm conditions directly to the rows of Y, we use the introduced matrix variable Y_{plus} to generate the following constraints:

$$\mathcal{L}_2(Y_{\text{plus}}) \leqslant X\,\mathcal{L}_2(B_{\text{plus}}) \qquad \text{and} \qquad \mathcal{L}_2((Y_{\text{plus}} - Y)_{\text{off}}) \leqslant X\,\mathcal{L}_2((B_{\text{plus}} - B)_{\text{off}}). \tag{6.41}$$

The following result shows that the new constraints in (6.40) and (6.41) are superior to the conditions in (6.39).

Lemma 6.5. *Consider a matrix triple* $(X, Y, Y_{\text{plus}}) \in \mathcal{D}^n \times \mathcal{S}^n \times \{\mathcal{S}^n \cap \mathcal{N}^n\}$*, and assume that the matrices* Y *and* Y_{plus} *satisfy the equalities*

$$\operatorname{diag}(Y) = X \operatorname{diag}(B), \quad Ye = XBe, \quad \operatorname{diag}(Y_{\text{plus}}) \equiv 0. \tag{6.42}$$

Then (6.40) *implies the validity of the vector constraint* $\mathcal{L}_1(Y) \leqslant X\,\mathcal{L}_1(B)$. *Furthermore,* (6.40) *and* (6.41) *together necessitate the compliance with the* 2*-norm inequalities* $\mathcal{L}_2(Y) \leqslant X\,\mathcal{L}_2(B)$.

Proof. By $Y_{\text{plus}} \geqslant_{\text{tri}} Y$ and $Y_{\text{plus}} \geqslant 0$, it follows $2Y_{\text{plus}} - Y_{\text{off}} \geqslant |Y_{\text{off}}|$. Together with the identity $|B| = 2B_{\text{plus}} - B_{\text{off}} + |B_{\text{diag}}|$, this yields

$$\begin{aligned}\mathcal{L}_1(Y) &= |Y_{\text{off}}|e + |\operatorname{diag}(Y)| \\ &\leqslant (2Y_{\text{plus}} - Y_{\text{off}})e + |\operatorname{diag}(Y)| \\ &= X(2B_{\text{plus}} - B_{\text{off}})e + |X\operatorname{diag}(B)| \\ &\leqslant X(2B_{\text{plus}} - B_{\text{off}} + |B_{\text{diag}}|)e \\ &= X\,\mathcal{L}_1(B)\end{aligned}$$

and verifies the first statement.

For the proof of the second statement, we use Lemma 2.10 and the convexity of norm functions. Together with the nonnegativity of $Y_{\text{plus}} \circ (Y_{\text{plus}} - Y_{\text{off}})$ and the identities $Y_{\text{plus}} \circ Y_{\text{diag}} = Y_{\text{off}} \circ Y_{\text{diag}} \equiv 0$, we get

$$\begin{aligned}\mathcal{L}_2(Y) &\leqslant \mathcal{L}_2([Y_{\text{plus}} \;\; Y_{\text{plus}} - Y_{\text{off}} \;\; Y_{\text{diag}}]) \\ &= \mathcal{L}_2([\mathcal{L}_2(Y_{\text{plus}}) \;\; \mathcal{L}_2(Y_{\text{plus}} - Y_{\text{off}}) \;\; \mathcal{L}_2(Y_{\text{diag}})]) \\ &\leqslant \mathcal{L}_2(X\,[\mathcal{L}_2(B_{\text{plus}}) \;\; \mathcal{L}_2(B_{\text{plus}} - B_{\text{off}}) \;\; \mathcal{L}_2(B_{\text{diag}})]) \\ &\leqslant X\,\mathcal{L}_2([\mathcal{L}_2(B_{\text{plus}}) \;\; \mathcal{L}_2(B_{\text{plus}} - B_{\text{off}}) \;\; \mathcal{L}_2(B_{\text{diag}})]).\end{aligned}$$

The last inequality is due to the convexity of $\mathcal{L}_2$ and the circumstance that X is doubly stochastic. Every row of X therefore describes a different convex combination of the rows of the matrix $[\mathcal{L}_2(B_{\text{plus}}) \;\; \mathcal{L}_2(B_{\text{plus}} - B_{\text{off}}) \;\; \mathcal{L}_2(B_{\text{diag}})]$. Finally, $B_{\text{plus}} \circ (B_{\text{plus}} - B_{\text{off}}) \equiv 0$ and $B_{\text{plus}} \circ B_{\text{diag}} = B_{\text{off}} \circ B_{\text{diag}} \equiv 0$ give the identity

$$\mathcal{L}_2([\mathcal{L}_2(B_{\text{plus}}) \;\; \mathcal{L}_2(B_{\text{plus}} - B_{\text{off}}) \;\; \mathcal{L}_2(B_{\text{diag}})]) = \mathcal{L}_2(B),$$

which completes the proof. □

6.2. Level-2 relaxations

With all the previously discussed constraints in our repertoire, we are now in the position to *finalize* the considered relaxation frameworks. This section serves the purpose of giving appropriate selections of constraints for the different types of relaxation frameworks. The following choice of cuts is driven by a trade-off between relaxation quality and efficiency.

We investigate the suitability of the respective constraints in the reversed order they were introduced, starting with the norm cuts presented in the last subsection and finishing with the *GLB* based cuts discussed in the first part of this section. Via numerical tests, it is easily shown that none of the presented norm constraints is redundant in any of the relaxations. Our improved version defined in (6.40) and (6.41) still gives a good trade-off between computational complexity and tightening effect on the relaxations. The application of the corresponding norm cuts in the level-2 versions of the respective relaxation is therefore strongly recommended. For the incorporation of these cuts, we introduce $\binom{n}{2}$ nonnegative variables to characterize a matrix variable $Y_{\text{plus}} \in \{M \in \mathcal{S}^n \cap \mathcal{N}^n \mid \operatorname{diag}(M) \equiv 0\}$ and add the constraints

$$\begin{aligned} Y_{\text{plus}} e &= X B_{\text{plus}} e, \\ Y &\leqslant_{\text{tri}} Y_{\text{plus}}, \\ \mathcal{L}_2(Y_{\text{plus}}) &\leqslant X\, \mathcal{L}_2(B_{\text{plus}}), \\ \mathcal{L}_2((Y_{\text{plus}} - Y)_{\text{off}}) &\leqslant X\, \mathcal{L}_2((B_{\text{plus}} - B)_{\text{off}}). \end{aligned} \tag{6.43}$$

The described procedure is the same for each relaxation framework. For reasons of efficiency, it is advisable to omit the corresponding second order cone inequalities in (6.43) whenever the applied solver has no explicit support for the SOCP constraints.[3]

Naturally, the relative tightening effect of the constraints in (6.43) depends on the quality of the relaxation which they have been added to. Independent of the considered relaxation framework and independent of the constellation of the other cuts, we made the following observation: except for a small number of QAP instances with very special coefficient structures, the conditions in (6.43) typically tighten the respective relaxations noticeably.

The second type of cuts that unites all these favorable properties is the class of sum-matrix bounds discussed in Subsection 6.1.3. In order to obtain strong cuts, we

[3]Since it is possible to formulate SOCP constraints via semidefinite conditions, every SDP solver has implicit support for this cone.

solve the auxiliary LP problem (6.32). From the computed solution vector, we obtain parameter vectors v_l, w_l, v_u, and w_u for the implementation of the following inequality constraints

$$Xv_le^T + ew_l^TX^T \leqslant_{\text{off}} Y \leqslant_{\text{off}} Xv_ue^T + ew_u^TX^T. \tag{6.44}$$

Using (6.44) often leads to a significant improvement of the relaxation quality. On top of this, the involved increase of computational costs is fairly moderate. In the context of these beneficial properties, it is clear that we apply (6.44) to the level-2 versions of each relaxation framework.

It is true that the nonnegativity condition $\Upsilon \geqslant 0$ can result in significantly tighter relaxation instances of VL_2. Nevertheless, we refrain from the integration of these conditions, since their use limits the practical applicability drastically. For dimensions $n > 30$, even in the absence of these constraints, *VL* based relaxations involve extremely high computational costs. With these inequalities being applied, it becomes very difficult to compute bounds even for problems of dimension $n \geqslant 20$. The situation is significantly better if the used solver handles the cone $\mathcal{S}_+ \cap \mathcal{N}$ efficiently. To the best of the author's knowledge, SDPNAL+ by Yang, Sun, and Toh [113] is the only solver that is capable of this. However, even in conjunction with this solver, vector lifting relaxations for QAPs of dimensions $n > 35$ become extremely expensive. On the contrary, it may be beneficial to reduce the (possibly) non-symmetric sum-matrix bounds in (6.44) to their symmetric counterparts if the computational cost plays a very critical role.

By Lemma 6.2, it is clear that the integration of *EVB* based cuts into the frameworks *ESC*, *ES*, or *VL* is typically redundant. This circumstance can be changed via a different QAP reformulation or the incorporation of upper bounds such as (6.25) and (6.26). Both types of modifications involve a significant increase of the computational costs and usually accompany hardly noticeable quality improvements of the respective relaxation. For this reason, we only consider the integration of *EVB* based cuts into the relaxations *ML*, *MLX*, *MS*, and *IIMS*. Under the assumptions that the QAP reformulation defined in (4.20) was applied a priori, it follows the existence of a specific eigenvalue-eigenvector pair $(\mu_1, p_1) = (\lambda_1, q_1) = (0, \frac{1}{\sqrt{n}}e)$. For the remaining eigenvalues of A, we assume that they are indexed in non-ascending order $\mu_2 \geqslant \mu_3 \geqslant \ldots \geqslant \mu_n$. Reversely, the eigenvalues of B shall be denoted in non-descending order $\lambda_2 \leqslant \lambda_3 \leqslant \ldots \leqslant \lambda_n$. These are the same presuppositions we made in (6.16). The incorporation of the *EVB* based cuts is realized

by adding the constraints

$$\begin{aligned} 0 &\leqslant \sum_{i=3}^{l} p_i^T Y p_i - \lambda_i \qquad \text{for} \quad l = 3, 5, 7, \ldots, 2\left\lfloor \frac{n}{2} \right\rfloor - 1, \\ (\lambda_{l+1} - \lambda_l)(1 - \bar{s}_{ll}) &\leqslant \sum_{i=2}^{l} p_i^T Y p_i - \lambda_i \qquad \text{for} \quad l = 2, 4, 6, \ldots, 2\left\lfloor \frac{n-1}{2} \right\rfloor, \end{aligned} \tag{6.45}$$

where $\bar{s}_{ll}$ is used as a replacement character for the linear term

$$\bar{s}_{ll} := \max\{u_{ll}, -l_{ll}\}\, |p_l|^T X |q_l| + \max\{u_{ll}\delta_{ll}^u, l_{ll}\delta_{ll}^l\}$$

with

$$l_{ll} = \langle p_l^{\downarrow}, q_l^{\uparrow} \rangle, \quad \delta_{ll}^l = l_{ll} + \langle |p_l^{\downarrow}|, |q_l^{\uparrow}| \rangle, \quad u_{ll} = \langle p_l^{\downarrow}, q_l^{\downarrow} \rangle, \quad \delta_{ll}^u = u_{ll} - \langle |p_l^{\downarrow}|, |q_l^{\downarrow}| \rangle.$$

For the integration into framework MS_2, we additionally exploit the observation made in Lemma 6.3. This means that we split and reduce the constraints in (6.45) by the same approach that was used for (6.14).

In the author's opinion, the incorporation of *GLB* based cuts is worth to be discussed. The addition of (6.8) or (6.9), respectively, is not redundant and affects the overall computational complexity only slightly. However, numerical tests have shown that - in the presence of the other discussed constraints - the influence of *GLB* based cuts on the bounding quality of the respective relaxation is even smaller than their slight effect on the computing times; typically, there is no effect at all. Another issue is the computation of suitable reformulation parameters. Of course it is possible to apply the reformulation procedures proposed by Assad and Xu [3] or Carraresi and Mallucelli [20] to improve the quality of the Gilmore-Lawler bound, which also strengthens the incorporated constraints. This, however, increases the computational costs significantly.

In the absence of the constraints (6.44) the application of *GLB* based cuts is still strongly recommended. Nevertheless, for the numerical examples presented in Section 7.4, we refrain from using these constraints and stay with the constellation of cuts given in (6.43), (6.44), and (6.45).

Chapter 7.

Implementation and Numerical Results

7.1. Motivation

In the previous chapters, we described different SDP relaxations for the QAP. We also discussed a range of applicable constraints in terms of their effect on the bounding quality and the accompanied computational costs. The former part has been emphasized to some extent by giving a few numerical examples. Regarding the latter part, however, the author clearly avoided the presentation or any further discussion of computing times. This section shall be used to give reason for this circumstance. To be more specific, the object of this section is to demonstrate the significant influence of different implementation strategies on the computing times as well as the numerical stability of the corresponding SDP frameworks.

In his research, the author had to realize that details about the actual implementation and their influence on the performance of the respective programming instances receive very little or no consideration in the majority of cited works that are concerned with convex relaxations of the QAP. Besides the typical distinction between the mathematical model and the technical realization, there is another important reason for this. In addition to high-level modeling languages for mathematical programming, such as AIMMS, AMPL, GAMS, or OPL, there exists a wide range of user-friendly interfaces to various conic optimization solvers for many different programming languages. The packages CVX [41], YALMIP [64], ROME [39], `Convex.jl` [104], and CVXPY [26] picture just a small selection of actively developed modeling tools for the programming languages MATLAB,

Julia, and Python. The recent improvements in stability and efficiency, as well as the continuously improving reformulation automatisms that have been incorporated into the listed modeling tools, allow the implementers to neglect various implementation details.

Despite the tremendous advances in the development of solvers and modeling languages, the consideration of certain implementation questions is still of importance for the design of numerically stable, memory efficient, and practically solvable programming problems. This circumstance shall be demonstrated via the numerical results presented in Table 7.1.

The accuracy of the computed approximations is measured by relative duality gaps. Let $\hat{\varrho}$ be the optimal objective value to a given SDP problem with a finite solution and zero duality gap.[1] The relative duality gap for an approximation is then defined as

$$d_{\text{rel}} = \frac{\tilde{\varrho}_{\text{prim}} - \tilde{\varrho}_{\text{dual}}}{|\hat{\varrho}|}, \tag{7.1}$$

where $\tilde{\varrho}_{\text{prim}}$ and $\tilde{\varrho}_{\text{dual}}$ are the computed primal and dual optimal objective values, respectively. In Table 7.1, we consider two different implementations *I1* and *I2*. The measurements that are labeled by the superscript ^{I1} correspond to a straightforward implementation of the corresponding relaxation instances, whereas the superscript ^{I2} refers to numerical results computed with an alternative implementation whose realization details are discussed in the following sections. The considered relaxation frameworks are problem (3.5) and problem (3.10), alternatively referred to as ML_0 and MS_0, respectively. The relations between the corresponding computing times are displayed in the fourth and seventh column of Table 7.1. Both implementations are realized using YALMIP [64]. The applied solver is SDPT3 [102] and the script language Octave [29].

Since *I1* and *I2* are just different realizations of the same relaxation, it is possible to reconstruct the same optimal points $(\hat{X}, \hat{Y}, \ldots)$ from the solution vectors to each implementation. The described feasible sets are indeed completely identical. In that context, it is surprising how much the displayed results differ between these two implementation strategies. The examples for *I2* perform considerably better than their *I1* counterparts. Particularly interesting are the QAP instances *Kra32*, *Tai35a*, and *Tho40*. For the *I1* implementations of the corresponding ML_0 instances, SDPT3 fails to compute feasible points (hence the large gaps). It is also quite difficult to evaluate the numerical accuracy of the other instances of *I1* since most of the computed objective values have negative duality gaps. The latter statement implies that the approximate primal or dual solution

[1] It can be shown that this is the case for every presented SDP relaxation.

Table 7.1.: Relative gaps and timings for different implementations and selected QAPs

	ML_0			MS_0		
Problem	d_{rel}^{I1}	d_{rel}^{I2}	t^{I1}/t^{I2}	d_{rel}^{I1}	d_{rel}^{I2}	t^{I1}/t^{I2}
Esc32g	$-2.27 \cdot 10^{-9}$	$7.02 \cdot 10^{-8}$	4.82	$-1.66 \cdot 10^{-9}$	$2.66 \cdot 10^{-8}$	5.79
Kra32	$-6.34 \cdot 10^{+3}$	$5.94 \cdot 10^{-6}$	16.88	$1.65 \cdot 10^{-7}$	$1.07 \cdot 10^{-7}$	4.48
LiPa40a	$-3.37 \cdot 10^{-4}$	$2.23 \cdot 10^{-6}$	8.05	$-8.64 \cdot 10^{-8}$	$5.44 \cdot 10^{-8}$	4.92
Nug30	$-1.79 \cdot 10^{-9}$	$5.43 \cdot 10^{-6}$	2.78	$-1.15 \cdot 10^{-9}$	$8.33 \cdot 10^{-6}$	3.66
Ste36a	$-5.51 \cdot 10^{-5}$	$1.66 \cdot 10^{-3}$	7.67	$5.18 \cdot 10^{-8}$	$8.82 \cdot 10^{-7}$	5.18
Tai35a	$-2.22 \cdot 10^{+3}$	$5.51 \cdot 10^{-7}$	9.80	$-1.07 \cdot 10^{-6}$	$1.44 \cdot 10^{-7}$	5.79
Tho40	$-8.06 \cdot 10^{+4}$	$7.69 \cdot 10^{-6}$	15.44	$-2.06 \cdot 10^{-8}$	$4.77 \cdot 10^{-6}$	4.78

violates certain constraints. On top of this, the computing times for the corresponding *I2* implementations are significantly shorter than the times required for solving their *I1* counterparts.

Similar numerical issues were observed in [83] by Peng, Mittelmann, and Li. The authors of [83] attacked this problem using a procedure described in [54]. Here the issue is resolved by applying the reformulation technique that will be described in the next section. Jansson's procedure for the computation of rigorous lower bounds [53, 54] is of course still applicable.

7.2. Numerical difficulties

Alongside the conic solvers SeDuMi [101], SDPT3 [102, 103], SDPA [112], SDPNAL+ [113, 117], and SCS [77], there are many more solvers which are capable of handling semidefinite programming problems. The use of such a solver reduces the implementation effort to the task of reformulating the given problem instance for compliance with the corresponding input format. The most important aspect of each input format is the underlying standard form. Fortunately, for almost all of these solvers, the required formulation can be traced back to the same SDP form. The *primal standard form* is

given as

$$\begin{aligned} \inf_{X \in \mathcal{S}^n_+} \quad & \langle \mathrm{C}, X \rangle \\ \text{s.t.} \quad & \langle \mathrm{A}_i, X \rangle = \mathrm{b}_i \qquad \text{for} \quad i = 1, \ldots, m, \end{aligned} \tag{PSDP}$$

with its dual

$$\begin{aligned} \sup_{y \in \mathbb{R}^m} \quad & \mathrm{b}^T y \\ \text{s.t.} \quad & \mathrm{C} - \sum_{i=1}^{m} y_i \mathrm{A}_i \in \mathcal{S}^n_+, \end{aligned} \tag{DSDP}$$

where $\mathrm{b} = (\mathrm{b}_1, \ldots, \mathrm{b}_m)^T \in \mathbb{R}^m$ and $\mathrm{A}_1, \ldots, \mathrm{A}_m, \mathrm{C} \in \mathcal{S}^n$.

If we attempt to use one of the listed solvers for computing the optima to our relaxation programs, it is reasonable to discuss numerical stability issues and implementation details in consideration of (PSDP) and (DSDP). Many of the following explanations will therefore refer to these standard forms.

Another important background for the following discussion is the notion of ill-posedness (well-posedness) in the context of semidefinite programming problems. To the best of the author's knowledge there are two definitions for condition measures of a linear programming problem. The first one was introduced by Mangasarian [71]. His definition of a condition number for a system of linear inequalities and equalities is related to the error bounds introduced by Robinson [90]. Robinson's results, in turn, are based on a well-known theorem by Hoffman [47]. For more recent results on this topic, see [8, 115] and the references therein. The second definition is due to Renegar [89], who approaches this topic from a slightly different perspective. He defines a condition measure with regard to the decision problem about the consistency of a system of constraints. Both types of condition measures lead to a very similar but not identical notion of ill-posedness. In the following, we consider the notion of *ill-posedness in regard to its decision problem* which was given by Renegar [89].

A problem instance in primal standard form is called ill-posed in regard to its decision problem if it lies in the intersection of the closure of feasible and infeasible problems of (PSDP). This is the case if at least one of the following three statements applies:

- The equality constraints on X are linearly dependent.
- The problem is weakly feasible, or

- weakly infeasible.

In the following, whenever a problem is characterized as *ill-posed*, the term is meant in regard to the decision problem about its consistency. The complement to the set of ill-posed problem instances is the set of well-posed ones. This includes not only strongly feasible problems whose linear map defined by $\{\mathrm{A}_i\}$ is onto, but also strongly infeasible optimization problems.

7.2.1. Ill-posed programming problems

The ill-posedness of an SDP problem accompanies a list of undesirable properties. The presence of redundant variables and constraints has a negative effect on the computational costs and the stability of the corresponding problem instance. Even more problematic are the cases for which Slater's condition [99] is not satisfied. The absence of interior feasible points can have a significant impact on the convergence rate in the solving procedure. Both circumstances are also major obstacles for the computation of verified bounds.

Fortunately, in the context of the discussed semidefinite programming relaxations, it is not necessary to consider weak infeasibility. The boundedness of each optimization variable implies the absence of weak infeasibility (c.f. [68, Proposition 3]). On the other hand, weak feasibility as well as linearly dependent equality constraints actually occur.
Theorem 7.1. *If the corresponding approximation tolerance ε is set to zero, every possible instance of any relaxation framework given in Chapter 3 - 5 is ill-posed.*

Proof. By construction and the nature of the underlying QAP, all discussed SDP relaxations necessarily state feasible programming problems. For the proof of Theorem 7.1, it is therefore sufficient to validate the absence of interior feasible solutions.

In the first part of the argument, we consider the basic relaxations presented in Section 3.1. The equality constraints (3.3c) and (3.3e) of the relaxation VL_0 imply the identity

$$\begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \begin{bmatrix} -e^T & -e^T \\ I \otimes e & e \otimes I \end{bmatrix} \equiv 0 \tag{7.2}$$

for every feasible point $(x, \mathcal{Y})$ to problem (3.3). By the same operandi, we use (3.5d), (3.10d), and (3.11d) to verify

$$\begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \begin{bmatrix} Be & e \\ 0 & -e \\ -e & 0 \end{bmatrix} \equiv 0, \tag{7.3}$$

$$\begin{bmatrix} B_\diamond & B_\diamond X^T \\ XB_\diamond & Y_\diamond \end{bmatrix} \begin{pmatrix} e \\ -e \end{pmatrix} \equiv 0 \qquad \text{for} \quad \diamond \in \{+,-\}, \tag{7.4}$$

and

$$\begin{bmatrix} 1 & q_i^T X^T \\ Xq_i & Q_i \end{bmatrix} \begin{pmatrix} q_i^T e \\ -e \end{pmatrix} \equiv 0 \qquad \text{for} \quad i = 1, \ldots, n, \tag{7.5}$$

respectively. Due to the rank deficiency within the accounted SDP constraints there are no feasible solutions that satisfy strict positive definiteness. The corresponding programs are weakly feasible because all feasible points lie on the boundary of the cone of semidefinite matrices.

For the remainder of the proof, we notice that any feasible programming problem which contains conditions implying the positive semidefiniteness of a variable $Z \in \mathcal{S}^n$ together with an equality constraint of the form

$$\langle E, Z \rangle = 0 \quad \Longleftrightarrow \quad e^T Z e = 0 \tag{7.6}$$

is necessarily ill-posed. The validity of this statement is an immediate consequence of the fact that arbitrary small negative perturbations of the right-hand side of (7.6) render this problem infeasible. Apparently, this argument is applicable to all level-1 and level-2 relaxations. The considered variables are $G_j, Q_i, U_{i\star}, Y_+, Y_-, Y_\triangle, Y_\triangledown$, and Z, respectively. □

The fact reported in Theorem 7.1 seems to be in stark contrast to statements about generic properties of SDP problems in [2, 28, 82]. Alizadeh, Haeberly, and Overton [2] showed that primal and dual nondegeneracy are generic properties of semidefinite programs. In [82], Pataki and Tunçel proved the corresponding generalization for conic linear programming problems. Recently, Dür, Jargalsaikhan, and Still [28] surveyed

different genericity results for this kind of programming problems. They showed that strong duality holds generically in a stronger sense. By taking these results into account, it is straightforward to show that well-posedness is a generic property of (PSDP) and (DSDP). Roughly speaking, this means that almost all linear semidefinite programming problems are well-posed.

The last statement about the genericity of well-posedness provokes an interesting question: How is it explainable that all derived relaxation programs are ill-posed, although the underlying relaxation techniques differ in various ways? This matter is particularly interesting considering the fact that the ill-posedness property is not limited to some specific implementation. The applied QAP reformulation introduces a constant nullspace ($Be \equiv 0$) which is present in all previously considered relaxation frameworks. Nevertheless, even if we remove any nullspace from the reformulated matrix B, the corresponding relaxation instance remains ill-posed.

Actually, the answer to the question above is quite simple: the design of relaxation frameworks is not haphazard. Our desire to model tight SDP relaxations often introduces redundancies and produces ill-posed programming problems. Indeed, the situation is very similar for a wide range of problems occurring in practice. The results in [2, 28, 82] hold for programming problems with specific conic structures but without consideration of any sparsity structure. They are therefore not applicable to the limited set of SDP problems that occur in practice. A good demonstration for the coherence of this statement is due to Ordóñez and Freund who found that 71% of the LP problem instances from the NETLIB test suite [76] have infinite condition measure. Numerical difficulties associated with the ill-posedness of these problems were investigated, for instance, by Keil and Jansson [58].

7.2.2. Regularization

An appropriate way to deal with ill-posed or strongly ill-conditioned problems is the application of regularization techniques. Subsequently, we will explain a possible procedure for the regularization of an ill-posed primal semidefinite programming problem. For this

purpose, consider the following modification of (PSDP):

$$\begin{aligned} \inf_{X\in\mathcal{S}^n,\,\rho\in\mathbb{R}_+} \quad & \langle \mathrm{C}, X\rangle \\ \text{s.t.} \quad & \langle \mathrm{A}_i, X\rangle = \mathrm{b}_i \qquad \text{for} \quad i = 1,\ldots,m, \\ & X + \rho I \in \mathcal{S}^n_+. \end{aligned} \tag{7.7}$$

Under the assumption that the linear map described via the coefficient matrices $\{\mathrm{A}_i\}$ is onto, the primal of the modified programming problem is evidently well-posed. However, without any further constraints on ρ or a penalization of nonzero values, the semidefiniteness condition on X is effectively nullified.

The general idea for the presented regularization approach follows the concept of penalty methods. Instead of requiring strict compliance with all initial constraints, we allow arbitrarily large perturbations to certain ones. The constraint violations are kept small for the respective optimal points by incorporating a penalization term into the objective function. In the standard form, a possible relaxation of (PSDP) may look like

$$\begin{aligned} \inf_{\tilde{X}\in\mathcal{S}^{n+1}_+} \quad & \left\langle \begin{bmatrix} \mathrm{C} & 0_{(n,1)} \\ 0^T_{(n,1)} & \mathrm{c}_\rho - \mathrm{tr}(\mathrm{C}) \end{bmatrix}, \tilde{X} \right\rangle \\ \text{s.t.} \quad & \left\langle \begin{bmatrix} \mathrm{A}_i & 0_{(n,1)} \\ 0^T_{(n,1)} & -\mathrm{tr}(\mathrm{A}_i) \end{bmatrix}, \tilde{X} \right\rangle = \mathrm{b}_i \qquad \text{for} \quad i = 1,\ldots,m, \end{aligned} \tag{7.8}$$

where c_ρ is an appropriate penalization coefficient and $\tilde{X}$ can be interpreted as a substitute for $\overset{*}{\mathrm{diag}}(X + \rho I, \rho)$.

In the process of designing an SDP relaxation framework, the redundancy of certain equality constraints may not always be apparent. Consider, for instance, the condition $X \in \mathcal{D}^n$. In [25, 73, 83, 84] and many other papers, this condition is realized via n^2 nonnegative variables and $2n$ equalities: $X \in \mathcal{N}^n$ and $Xe = X^T e = e$. It is straightforward to show that one of the corresponding equality constraints is redundant. The lack of attention to this simple detail is already enough to render the problem ill-posed. In contrast to the given example, there are other redundancies whose detection and validation can be a comparably difficult task. The following relaxation deals with this by allowing

a perturbation of the right-hand side of the equality constraints:

$$\begin{aligned} \inf_{X \in \mathcal{S}^n_+,\, p \in \mathbb{R}^m} \quad & \langle \mathrm{C}, X \rangle + \mathrm{c}_p \|p\| \\ \text{s. t.} \quad & \langle \mathrm{A}_i, X \rangle = b_i + p_i \qquad \text{for} \quad i = 1, \ldots, m. \end{aligned} \tag{7.9}$$

For reasons of clarity, we skip the formulation of problem (7.9) in standard form. The same applies to the treatment of ill-posed dual problems. The general regularization strategy is very similar to the one for the primal SDP problem.

A closer look at the presented approaches reveals their connection to typical regularization techniques from which they inherit some very beneficial properties. In exchange for inexact results, we derive a well-posed relaxation that requires only marginal changes to the structure of the base model. The modifications increase the problem dimension only slightly and retain the sparsity structure of the coefficient matrices. They thereby have a relatively small footprint on the computational costs of the solving procedure. On top of that, the described approaches are applicable without knowledge about the constraints which cause the ill-posedness of the considered problem.

7.2.3. Minimal face representation

The regularization procedure is a good technique for the construction of well-posed approximations of the original programming problem. It can be realized straightforwardly and has many beneficial properties. Nevertheless, the procedure is not suitable as a general approach. Though the constructed approximation can be very strong, it is still just that, an approximation. It is therefore not applicable if we attempt to evaluate certain properties of the original relaxation program. It can also not be used to compute verified upper bounds or to determine infeasibility. The latter situation may be an issue if the relaxation is embedded in a branch-&-bound procedure. A sensible approach that resolves the ill-posedness but pertains the addressed properties requires certain reformulations.

Several years ago, in [44], Hadley, Rendl, and Wolkowicz proposed a relaxation model for the QAP which implicitly incorporates the requirement $X \in \mathcal{E}$ by a reformulation via projection. Although their stimulus was different from our current one, the considered procedure is exactly what is needed to encounter the issue shown in Theorem 7.1. The reformulation via projection implements a suggestion of Boyd who pointed out that:

> [...] any feasible nonstrict LMI can be reduced to an equivalent LMI that is strictly feasible, by eliminating implicit equality constraints and then reducing the resulting LMI by removing any constant nullspace. [12, Chapter 2.5.1]

Regarding the discussed relaxations for the quadratic assignment problem the culprit is easily found. The presence of a constant nullspace is originated in the condition $X \in \mathcal{E}$. To be more precise, the ill-posedness is caused by the constant subspace defined via the vector equality $X^T e = e$ and the quadratic nature of the respective semidefiniteness conditions.

A possible approach to address this issue is the reduction of the corresponding programming variables to a smaller subspace. By providing a tractable representation of the linear manifold spanned by $\mathcal{E}$, one can eliminate the explicit equality constraints $Xe = X^T e = e$ and resolve the accompanied ill-posedness. A good example of this procedure is given in [27]. Though Theorem 7.1 validates the existence of a constant nonempty nullspace in the feasible sets of $MSDR_1$ instances, this issue is resolved in the final version of this framework which is referred to as $MSDR_3$. In order to obtain stronger bounds, the authors of [27] apply their relaxation to a projected version of the quadratic assignment problem based on a reformulation by Hadley, Rendl, and Wolkowicz [44]:

$$\begin{array}{ll} \inf\limits_{\tilde{X}\in\mathcal{Q}^{n-1}} & \left\langle V^T AV\tilde{X}V^T BV + V^T(C + \frac{2}{n}AEB)V, \tilde{X}\right\rangle + \left\langle C + \frac{1}{n}AEB, \frac{1}{n}E\right\rangle \\ \text{s.t.} & V\tilde{X}V^T \geqslant -\frac{1}{n}E, \end{array} \tag{PQAP}$$

where V is a real $n \times (n-1)$ matrix that satisfies $V^T V = I_{(n-1)}$ and $V^T e \equiv 0$. It is straightforward to show that (PQAP) is indeed equivalent to (KBQAP). The corresponding optimization variables are related via

$$X = V\tilde{X}V^T + \frac{1}{n}E.$$

By applying their framework to the reformulated version (PQAP), Ding and Wolkowicz [27] not only tighten the relaxation, they also decrease the number of equality constraints and reduce the dimension of the semidefiniteness condition. A positive side effect of this reformulation procedure is the resolution of ill-posedness. After a closer look on the other presented relaxation frameworks, it becomes apparent that the same idea can be used to construct well-posed equivalents to all presented relaxations for the QAP.

The process of reducing an arbitrary SDP problem with no strict interior feasible points to an equivalent program in smaller space satisfying Slater's condition is called

minimal face reduction.[2] The development of this procedure goes back to the early 80's when Borwein and Wolkowicz worked on a Lagrange multiplier theorem which holds without any constraint qualification [10,11]. For more recent developments on this topic, see [22,23,67,81,106] and the references therein. In the context of the presented relaxation programs, it should also be mentioned that a detailed explanation about the design of a minimal face representation for the framework *VL* is given in [116] by Zhao, Karisch, Rendl, and Wolkowicz. The described procedure was also adopted in [87] by Rendl and Sotirov to ensure the existence of interior feasible solutions.

For a brief recap of the general reformulation procedure, consider the SDP problem given in primal standard form. Denote by $\mathcal{F}$ the feasible set to the considered instance of (PSDP). The corresponding nullspace - which Boyd is referring to in [12, Subsection 2.5.1] - can be defined as

$$\operatorname{null}(\mathcal{F}) := \{v \in \mathbb{R}^n \mid \forall X \in \mathcal{F}\colon Xv \equiv 0\}. \tag{7.10}$$

Let n_r be the dimension of the orthogonal complement of $\operatorname{null}(\mathcal{F})$, i.e. $n_r := \dim(\operatorname{null}(\mathcal{F})^\perp)$. Furthermore, define a matrix $W \in \mathcal{M}^{n_r,n}$ that satisfies

$$\forall v \in \operatorname{null}(\mathcal{F}),\ w \in \operatorname{null}(\mathcal{F})^\perp \backslash \{0\}\colon \quad Wv = 0 \neq Ww. \tag{7.11}$$

Apparently, W has full rank and its row vectors state a basis of $\operatorname{null}(\mathcal{F})^\perp$. As an immediate consequence of (7.10) and (7.11), we have

$$\forall X \in \mathcal{F}\colon \quad XW^\dagger W = X = W^\dagger W X = (W^\dagger W)^T X.$$

For every $X \in \mathcal{F}$, there exists an $\tilde{n} \times \tilde{n}$ symmetric positive semidefinite matrix $\tilde{X}$ satisfying the two identities $\tilde{X} = W^{\dagger T} X W^\dagger$ and $X = W^T \tilde{X} W$. The latter identity is used to reformulate (PSDP) in the following form:

$$\begin{array}{ll} \inf\limits_{\tilde{X} \in \mathcal{S}^{\tilde{n}}_+} & \left\langle W \mathrm{C} W^T, \tilde{X} \right\rangle \\ \text{s.t.} & \left\langle W \mathrm{A}_i W^T, \tilde{X} \right\rangle = \mathrm{b}_i \qquad \text{for} \quad i = 1, \ldots, m. \end{array} \tag{7.12}$$

By (7.10) and (7.11), we deduce the existence of some strictly interior feasible $\tilde{X} \in \mathcal{S}^k_{++}$ and validate the conformance with Slater's condition. The final step of the reformulation

[2] This description actually refers only to a specific version of minimal face reduction procedures. The generalization of this reduction scheme can be applied to arbitrary conic linear programming problems.

procedure requires the elimination of all redundant equality constraints. We therefore reduce the coefficient matrices $\{W\mathrm{A}_iW^T\}$ to a linearly independent base.

For reasons of comprehensibility, let us discuss an actual example for the minimal face reduction of MS_0. The different LP and SDP constraints of problem (3.10) result in a block-diagonal form of the optimization variables of the corresponding (PSDP) instance. One can exploit this block-diagonal structure by regarding the respective nullspace for each block separately. In the case of problem (3.10), it is sufficient to limit our consideration to the SDP constraints in (3.10b). The relevant, parameter independent parts of the nullspaces have already been used in (7.4) for the proof of Theorem 7.1. Additionally, the nullspaces depend on the kernel of B_+ and B_-, respectively. We denote the feasible sets for each SDP block individually by $\mathcal{F}_+$ and $\mathcal{F}_-$. The accompanied nullspaces are

$$\mathrm{null}(\mathcal{F}_\diamond) = \left\{ \begin{pmatrix} v - \alpha e \\ \alpha e \end{pmatrix} \;\middle|\; \alpha \in \mathbb{R},\, v \in \mathbb{R}^n,\, B_\diamond v \equiv 0 \right\} \qquad \text{for} \quad \diamond \in \{+,-\}.$$

We define the corresponding transformation matrices for the minimal face reduction in regard to a matrix $V \in \{M \in \mathcal{Q}^{n,n-1} \mid M^T e \equiv 0\}$:

$$W_\diamond := \begin{bmatrix} Q_{\Omega^0_\diamond} & 0_{(n,n-1)} \\ \frac{1}{n} E Q_{\Omega^0_\diamond} & V \end{bmatrix}^T \qquad \text{for} \quad \diamond \in \{+,-\},$$

where the index set definitions for Ω^0_+, Ω^0_- are taken from (5.4). The usage of these index set subscripts is described in (5.5). By the compliance with all equality constraints of problem (3.10), one can easily validate that

$$\forall \diamond \in \{+,-\}\colon \quad W_\diamond^{\dagger T} \begin{bmatrix} B_\diamond & B_\diamond X^T \\ X B_\diamond & Y_\diamond \end{bmatrix} W_\diamond^\dagger = \begin{bmatrix} \Lambda_{\Omega^0_\diamond} & Q^T_{\Omega^0_\diamond} B_\diamond X^T V \\ V^T X B_\diamond Q_{\Omega^0_\diamond} & V^T Y_\diamond V \end{bmatrix}.$$

We exploit these identities to construct the following reformulation of MS_0:

$$\inf_{X\in\mathcal{D}^n,\ \tilde{Y}_+,\tilde{Y}_-\in\mathcal{S}^{n-1}} \quad \langle A, \mathbf{Y}(X, \tilde{Y}_+ - \tilde{Y}_-, B)\rangle + \langle C, X\rangle \tag{7.13a}$$

$$\text{s. t.} \quad \begin{bmatrix} \Lambda_{\Omega^0_\diamond} & Q^T_{\Omega^0_\diamond} B_\diamond X^T V \\ V^T X B_\diamond Q_{\Omega^0_\diamond} & \tilde{Y}_\diamond \end{bmatrix} \in \mathcal{S}_+, \qquad \diamond \in \{+,-\}, \tag{7.13b}$$

$$\operatorname{diag}(\mathbf{Y}(X, \tilde{Y}_\diamond, B_\diamond)) = X \operatorname{diag}(B_\diamond), \qquad \diamond \in \{+,-\}, \tag{7.13c}$$

where $\{\tilde{Y}_\diamond\}$ are used as substitutes for $\{V^T Y_\diamond V\}$, and the transformation $\mathbf{Y}\colon \mathcal{D}^n \times \mathcal{S}^{n-1} \times \mathcal{S}^n \to \mathcal{S}^n$ is given by

$$\mathbf{Y}(X, \tilde{Y}_\diamond, B_\diamond) = V \tilde{Y}_\diamond V^T + \frac{1}{n}\left(E B_\diamond X^T + V V^T X B_\diamond E\right). \tag{7.14}$$

Since $\mathbf{Y}(X, \tilde{Y}_\diamond, B_\diamond)e = X B_\diamond e$ holds valid for every $X \in \mathcal{D}^n$, the explicit incorporation of these equality constraints is redundant.

If one of the matrices B_+, B_- is representable as a sum-matrix, then the respective semidefiniteness condition is irrelevant. In all other cases, problem (7.13) contains interior feasible points, such as

$$\left(X, \tilde{Y}_+, \tilde{Y}_-\right) = \left(\frac{1}{n}E,\ \frac{\langle B_+, VV^T\rangle}{n-1} I_{(n-1)},\ \frac{\langle B_-, VV^T\rangle}{n-1} I_{(n-1)}\right).$$

The problem is well-posed if the accompanied equality constraints are linearly independent. In order to satisfy this requirement, the condition $X \in \mathcal{E}^n$ has to be implemented via $2n-1$ linearly independent equalities. These can be obtained as a subset of equality constraints from $Xe = X^T e = e$.

7.2.4. Remarks on applicability of regularizing procedures

Regarding their practical applicability, both the regularization approach presented in Subsection 7.2.2 and the minimal face reformulation given in Subsection 7.2.3 have their pitfalls. A beneficial utilization of the described techniques requires a good portion of cautiousness since a straightforward implementation of these may result in semidefinite programming problems which are even more difficult to solve with the available SDP solvers.

Two major drawbacks of the presented regularization approach have already been pointed out in the last subsection. The initial qualification as an "appropriate way to deal with ill-posedness" is indeed a strong exaggeration for this technique. Its application leads to a well-posed approximation with a slightly weaker bound. Properties like boundedness or infeasibility are not preserved. In that context, the regularization approach seems to be hardly appropriate for the computation of verified bounds or the application in a branch-&-bound procedure. In addition to these limitations, the regularization techniques involve the difficult task of finding appropriate penalty coefficients. Too small values cause weak bounds, whereas unnecessary large ones produce ill-conditioned objective functions. The former situation renders the relaxation unusable, the latter has a negative effect on the convergence rate and the overall numerical stability.

Fortunately, in the context of the discussed relaxation programs, we can overcome most of these issues. For the QAP, it is possible to implement a branch-&-bound algorithm in such a way that every branching node contains feasible solutions. The boundedness of all variables can be recovered by setting explicit bounds for the introduced constraint violations, i.e. $\rho \leqslant \text{const}$. Reasoned in our detailed knowledge about the respective problem structures, we are in the position to compute good estimations for appropriate penalty coefficients and reduce the constraint modifications to the necessary minimum.

The last statements shall be exemplified by the following redesign of ML_1:

$$\inf_{X\in\mathcal{D}^n,\,\tilde{Y},\tilde{Z}\in\mathcal{S}^n,\,p\in\mathbb{R}^2} \quad \langle A,\tilde{Y}\rangle + \langle C,X\rangle + \langle c_p,p\rangle \tag{7.15a}$$

$$\text{s. t.} \quad \begin{bmatrix} I & X^T & BX^T \\ X & I+p_1 I & \tilde{Y} \\ XB & \tilde{Y} & \tilde{Z} \end{bmatrix} \in \mathcal{S}_+, \tag{7.15b}$$

$$\operatorname{diag}(\tilde{Y}) = X\operatorname{diag}(B), \quad \operatorname{diag}(\tilde{Z}) = X\operatorname{diag}(B^2) + p_2 e, \tag{7.15c}$$

$$\langle I-E,\tilde{Z}\rangle = \langle I-E,B^2\rangle, \tag{7.15d}$$

$$p_1 \leqslant 1, \quad p_2 \leqslant \|B\|^2. \tag{7.15e}$$

This relaxation is derived from problem (4.31) by incorporating the defects $p_i I$ into the respective optimization variables. Positive defects are penalized in the objective function multiplied by the coefficients in vector $c_p \in \mathbb{R}^2$. In order to restore the boundedness of all variables, we added the inequalities in (7.15e). Negative coordinates in p are not feasible:

the semidefiniteness condition implies $\tilde{Z} \succeq XB^2X^T$ and thereby

$$\operatorname{tr}(\tilde{Z}) = e^T\tilde{Z}e + \langle I - E, \tilde{Z}\rangle \geqslant e^TXB^2X^Te + \langle I - E, \tilde{Z}\rangle = \operatorname{tr}(B^2).$$

Together with (7.15c), this necessitates $p_2 \geqslant 0$. By a very similar argument, $I + p_1 I \succeq XX^T$ implies the nonnegativity of p_1.

There are some details where the applied modifications in problem (7.15) deviate from the standard approach described in Subsection 7.2.2. The diagonal defect is subdivided into partitions to allow individual penalization of differently scaled matrix blocks. Defects are incorporated solely into the critical part of the semidefiniteness condition. Furthermore, by removing the redundant coefficients on the diagonal elements of $\tilde{Y}$ in (7.15d), we avoided unnecessary entries of perturbation variables p_i, which would have been present in the straightforward design: $\langle E, \tilde{Z}\rangle - np_2 = \langle E, B^2\rangle$.

For the determination of appropriate penalization coefficients in c_p, we first need to analyze the relation between the respective diagonal perturbations and possible perturbations ΔY of variable Y. Constraint (7.15b) implies

$$(1 + p_1)\tilde{Z} \succeq \tilde{Y}^2 = (Y + \Delta Y)^2 = Y^2 + (\Delta Y\, Y + Y\, \Delta Y + \Delta Y^2). \tag{7.16}$$

This relation gives a rough indication for the relative disposition of the respective optimal points: perturbations of Y involve perturbations of the left-hand side of (7.16) which are scaled by $2|||Y|||_2 \approx 2|||B|||_2$. For small values of p_1, this implies

$$p_2 \approx |||\tilde{Z} - Z|||_2 \gtrapprox |||\tilde{Z} - Y^2|||_2 \gtrapprox 2|||Y|||_2\, |||\Delta Y|||_2 \approx 2|||B|||_2\, |||\Delta Y|||_2.$$

Moreover, (7.16) and the assumption $|||\tilde{Z}|||_2 \approx |||B|||_2^2$ give

$$p_1 \gtrapprox |||B|||_2^{-1} |||\Delta Y|||_2.$$

The final ingredient used to obtain appropriate penalization coefficients is the estimate

$$|\langle A, \Delta Y\rangle| \leqslant |||A|||_{\mathrm{T}}\, |||\Delta Y|||_2.$$

For the goal of keeping the optimal objective value to the regularized problem very close to the original optimal objective value, it is necessary to request $\langle A, \Delta Y\rangle + \langle c_p, p\rangle \gtrapprox 0$. Based on this and the estimates above, we educe the following suggestions for the vector

of penalization coefficients:

$$c_p = \kappa \, |||A|||_{\mathrm{T}} \begin{pmatrix} 2|||B|||_2 \\ |||B|||_2^{-1} \end{pmatrix} \tag{7.17}$$

for some κ in the interval $\big[1,\ 50\big]$.

The applied modifications are fixing most of the discussed issues of the regularization approach from Subsection 7.2.2. Its applicability is nevertheless strongly limited. For feasibility tests and the computation of rigorous upper bounds, the approach is generally not applicable. On top of this, the partitioning of the introduced defect and the computation of suitable penalization coefficients require knowledge about specific modeling details of the considered problem.

The difficulties with the minimal face reduction approach are of different nature but existent nevertheless. In particular, this refers to the knowledge of the constant nullspace. It is possible to automatize the minimal face reduction procedure by solving stable auxiliary subproblems which serve the computation of (dual) recessing directions or a similar type of reduction directions. Corresponding algorithms are described in [23, 81] and [67]. However, the construction of a minimal face presentation by iterative computations of reduction directions can be very expensive. In the presence of rounding errors, it is moreover not possible to guarantee the equivalence of the computationally obtained presentations and the programs they originated from.

Even if the nullspace is given - which means that the minimal face reduction is easily applicable -, the reformulation may not help to resolve the numerical difficulties accompanied by the original SDP problem. It is actually possible that the obtained minimal face representation is more difficult to solve. An important factor in this context is the increased fill-in of the coefficient matrices. For an illustration, consider the diagonal equality constraints in (3.10c) and their counterparts for the reformulated relaxation framework given in (7.13c). Each of the original equality constraints described by the vector identities $\operatorname{diag}(Y_\diamond) = X \operatorname{diag}(B_\diamond)$ necessitates $n+1$ nonzero coefficients in the actual implementation. The number of coefficients that is necessary to implement (7.13c) depends on the choice of the matrix V. The application of the simple but dense

representative used in [44],

$$V = \begin{bmatrix} -\beta e_{(n-1)}^T \\ I_{(n-1)} - \alpha E_{(n-1)} \end{bmatrix} \quad \text{with} \quad \alpha = \frac{1}{n+\sqrt{n}},\ \beta = \frac{1}{\sqrt{n}},$$

leads to the introduction of $2n^2 - 2n + 1$ additional nonzero coefficients for each of the $2n$ equality constraints. If we replace the term $VV^T \mathcal{X} B_\diamond E$ in (7.14) with $\mathcal{X} B_\diamond E - \frac{1}{n} E B_\diamond E$, this number reduces to $n^2 - n + 1$ coefficients per equality. The additional fill-in is nevertheless immense.

In modern SDP solvers the sparsity structure of the coefficient matrices influences the computing times and the numerical stability. Roughly speaking, an increased number of nonzero coefficients due to constraints requiring linear transformations such as (7.14) may nullify the advantages of the reformulation approach described in Subsection 7.2.3. With the additional LP and SOCP constraints presented in Section 6.1 the situation becomes even worse. The heavily increased fill-in leads to poor convergence properties and higher computational costs.

In order to overcome this issue, one may search for sparser choices of V, as given by the following recursive definition:

$$V = \mathbf{V}(n) := \begin{cases} \begin{bmatrix} \mathbf{V}(n_1) & \gamma e_{(n_1)} & 0_{(n_1,n_2-1)} \\ 0_{(n_2,n_1-1)} & \delta e_{(n_2)} & \mathbf{V}(n_2) \end{bmatrix} & \text{if} \quad n > 1, \\ \begin{bmatrix}\ \end{bmatrix} & \text{else,} \end{cases} \tag{7.18}$$

where $n_1 = \left\lceil \frac{n}{2} \right\rceil$, $n_2 = n - n_1$, $\gamma = \sqrt{\frac{n_2}{n\,n_1}}$, and $\delta = -\sqrt{\frac{n_1}{n\,n_2}}$. This representative of V contains only about $n \log_2 n$ nonzero elements, which decreases the average fill-in per equality constraint to $\mathcal{O}(n \log_2 n)$.

For a given subspace, it can be very difficult to construct a sparse orthogonal basis. It is, for instance, not possible to give a construction rule for the matrix V such that the number of its nonzero elements is bounded by $\mathcal{O}(n)$. Fortunately, there is no substantial necessity for orthogonality. For a good reason, the transformation matrix W used in (7.11) is not required to have orthogonal row vectors. The essential requirement on W is that its rows span the orthogonal complement of the respective nullspace. If the dimension of this nullspace is significantly smaller than the dimension of its orthogonal complement,

then it is possible to construct transformation matrices W with very beneficial sparsity structures.

Let us once again consider the framework ML_1. The nullspace accompanied by the feasible set to the semidefinite block matrix in (4.31b) is already indicated in (7.3). The corresponding set of feasible semidefinite matrices shall be denoted by $\mathcal{F}$. The accompanied nullspace is given by

$$\text{null}(\mathcal{F}) = \{U^T v \mid v \in \mathbb{R}^2\} = \mathcal{R}(U^T) \qquad \text{with} \quad U = \begin{bmatrix} e & 0 \\ -e & 0 \\ 0 & -e \end{bmatrix}^T . \tag{7.19}$$

We are thus looking for a matrix $W \in \mathcal{M}^{3n-2,3n}$ that has full rank and satisfies $WU^T \equiv 0$. In addition, we demand a high sparsity level of W. For the construction of this matrix, it is sufficient to find an index set for a regular basis of column vectors of U. Let $\Psi := \{2n, 3n\}$ be this index set, and denote by Ψ' the complement to Ψ in $\{1, \ldots, 3n\}$, such that

$$U_\Psi = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \text{and} \qquad U_{\Psi'} = \begin{bmatrix} e_{(n)} & 0_{(n)} \\ -e_{(n-1)} & 0_{(n-1)} \\ 0_{(n-1)} & -e_{(n-1)} \end{bmatrix}^T .$$

The index sets Ψ and Ψ' can then be used to construct a sparse representative for W:

$$W \in \mathcal{M}^{3n-2,3n}: \qquad W_\Psi = (-U_\Psi^{-1} U_{\Psi'})^T, \quad W_{\Psi'} = I_{(3n-2)}. \tag{7.20}$$

Evidently, the matrix defined via (7.20) satisfies the basic condition $WU^T \equiv 0$ as well as the full rank requirement. On top of this, the specific sparsity structure of W has the beneficial side effect that the reduced SDP constraint in the corresponding minimal face representation is derived from the original semidefiniteness condition simply by removing the row and column vectors specified in Ψ. The sparsity structure is thus kept unaltered for all components of matrix variables that are not positioned on the respective rows and/or columns.

Based on these observations, we design the following minimal face representative for ML_1:

$$\inf_{\tilde{X}\in\mathcal{N}^{n-1,n},\ \tilde{Y},\tilde{Z}\in\mathcal{S}^{n-1}} \quad \langle A, \mathbf{Y}(\tilde{Y})\rangle + \langle C, \mathbf{X}(\tilde{X})\rangle \tag{7.21a}$$

$$\text{s. t.} \quad \begin{bmatrix} I_{(n)} & \tilde{X}^T & B\tilde{X}^T \\ \tilde{X} & I_{(n-1)} & \tilde{Y} \\ \tilde{X}B & \tilde{Y} & \tilde{Z} \end{bmatrix} \in \mathcal{S}_+^{3n-2}, \tag{7.21b}$$

$$\begin{aligned} &\operatorname{diag}(\tilde{Y}) = \tilde{X}\operatorname{diag}(B), \quad \operatorname{tr}(\mathbf{Y}(\tilde{Y})) = \operatorname{tr}(B), \\ &\operatorname{diag}(\tilde{Z}) = \tilde{X}\operatorname{diag}(B^2), \quad \operatorname{tr}(\mathbf{Y}(\tilde{Z})) = \operatorname{tr}(B^2), \end{aligned} \tag{7.21c}$$

$$\tilde{X}e_{(n)} \equiv 1, \quad \tilde{X}^T e_{(n-1)} \leqslant e_{(n)}, \tag{7.21d}$$

where the corresponding transformations are given as

$$\mathbf{X}(\tilde{X}) = \begin{bmatrix} \tilde{X} \\ e_{(n)}^T - e_{(n-1)}^T\tilde{X} \end{bmatrix} \qquad \text{and} \qquad \mathbf{Y}(\tilde{Y}) = \begin{bmatrix} \tilde{Y} & -\tilde{Y}e_{(n-1)} \\ -e_{(n-1)}^T\tilde{Y} & \langle E, \tilde{Y}\rangle \end{bmatrix}. \tag{7.21e}$$

The matrix variables X, Y, and Z have been modified with respect to the eliminated row and column vectors. For a further reduction of the number of nonzero coefficient entries, we replaced the equality constraints on the last elements of the diagonal vectors $\operatorname{diag}(\mathbf{Y}(\tilde{Y}))$ and $\operatorname{diag}(\mathbf{Y}(\tilde{Z}))$ with the respective trace equalities in (7.21c). The mutual implication of the replaced constraints follows immediately from the remaining diagonal equalities.

Numerical tests suggest that the described procedure is applicable in a practical manner and more favorable than the original problem formulation. There are still matters of detail that require individual treatment, such as the handling of eliminated variables which remain to be used implicitly, or the proper dealing with a possibly high condition measure of the reduced problem instance. Nevertheless, since these issues seem to be easily fixable for the individual problem instance, the author is convinced that the described version of the minimal face reduction is preferable to the regularization approach described in Subsection 7.2.2.

Considering the many additional constraints discussed in Section 6.1 that involve the use of $\mathbf{X}(\tilde{X})$ and $\mathbf{Y}(\tilde{Y})$, it can be advantageous to spent additional variables for the components that are not contained in the respective matrix variables $\tilde{X}$ and $\tilde{Y}$. This approach requires one additional equality constraint for each of the removed components.

On the other hand, the reduction of the overall fill-in of the coefficient matrices caused by the implicit use of the eliminated coordinates often lead to improved computing times.

Our final remark in this subsection concerns the condition measure of the respective problem instance. For a given SDP problem, the described minimal face reduction approach can be used to construct an equivalent problem which is well-posed and therefore suitable for the application of an interior point algorithm. By no means, however, does the exploitation of this procedure induce the creation of a well-conditioned programming problem. An unbalanced scaling of the respective programming variables can cause a bad convergence behavior and noticeable numerical problems. For the relaxation programs based on matrix lifting, the correlation $|||Z|||_2 \approx |||B|||_2 |||Y|||_2 \approx |||B^2|||_2 |||X|||_2$ indicates a particular sensitivity to the scaling of the respective QAP instance. To overcome the accompanied numerical difficulties, we normalize the parameter matrix B in such a way that the trace of B^2 is equal to its rank. To be precise, we determine the corresponding scaling factor

$$\alpha := \sqrt{\frac{|||B|||_{\mathrm{F}}^2}{\mathrm{rank}(B)}}. \tag{7.22}$$

and apply a counterbalanced scaling to the parameter matrices A and B, which creates a reformulated QAP instance $(\alpha A, \alpha^{-1} B, C)$.

7.2.5. Rigorous bounds via verification methods

The awareness about the presence of numerical instabilities leads us to the reasonable question whether it is possible to compute rigorous bounds for the true optimum of the respective relaxation framework. The answer to this question is “yes”, albeit with minor restrictions.

In 2004, Jansson introduced a new method to compute rigorous upper and lower bounds for the optimal objective values of linear programming problems [52]. The proposed procedure works much more efficient than previous approaches and is applicable even in the degenerated case. A short time later, the procedure was adapted for more general convex conic programming problems [53, 54] and implemented in the MATLAB software package VSDP [46]. In this subsection, we discuss an appropriate way to compute verified bounds for the optimal objective values to the respective QAP relaxations using the software package VSDP.

For the use of VSDP, it is required to install two further software packages: an SDP solver for computing approximate solutions to the given problem instances and the interval package INTLAB by Rump [93]. The latter is used mainly for the computation of rigorous inclusions of solutions to the accompanied linear systems [92, 94, 95]. Other important features of the package INTLAB that are relevant in this context are the support of interval input data in VSDP as well as the possibility to handle rounding errors in the computations which are required for the preparation of the respective relaxation instances.

In the software package VSDP, different post-processing routines are implemented for the result verification of semidefinite programming problems. Given the correct input data (see below) of an SDP program, rigorous bounds for the true optimal objective value can be computed. All rounding errors due to floating point arithmetic are taken into account. Since the necessary post-processing and the result verification is done within VSDP - and thereby do not require additional involvement of the user -, here we concentrate on the necessary preparation steps.

In order to obtain verified bounds for the optimal objective value to the original relaxation program, we must make sure that the input data for VSDP are correct. This means that the (interval) input data must contain the original problem. There are various ways to ensure this inclusion requirement; the choice depends on the specific relaxation model and the available verification tools. Though the use of INTLAB simplifies the implementation of the required computations significantly, a straightforward application of the available verification techniques is not recommended. This begins with the validation of the applied QAP reformulation defined in (4.20). The attempt to compute rigorous interval inclusions for the reformulation parameters in (4.19) seems quite naive. Firstly, with todays verification methods this attempt is difficult to realize for the solution to problem (4.18) and may fail due to degeneracy. Secondly, the early introduction of an interval matrix $\mathbf{B}$ results in additional data dependencies. These, in turn, lead to larger interval expansions. It is an important principle of verification methods to avoid such data dependencies, see [94, *Utilize Input Data Principle* (5.13)].

Another obstacle for the straightforward application of interval arithmetic is that most of the discussed SDP relaxations require an eigenvalue decomposition of the parameter matrix B. By exploitation of the symmetry of B and an appropriate application of Weyl's eigenvalue inequality [107], it is possible to compute rigorous inclusions for the eigenvalues of B very efficiently. On the other hand, the determination of rigorous

inclusions for all eigenvectors of B is more difficult, involves greater computational costs, and results in noticeable interval expansions.

Many practical verification methods make use of floating-point approximations and compute rigorous inclusions via implementation of a backward directed error handling. For this purpose, they often exploit suitable results from perturbation theory. Here we are following the same approach. Let $\tilde{Q}\tilde{\Lambda}\tilde{Q}^T$ denote an approximation of the eigenvalue decomposition of B, hence

$$B \approx \tilde{Q}\tilde{\Lambda}\tilde{Q}^T \approx \mathring{B} + \operatorname{diag}^*(\tilde{d}_B) + \tilde{v}_B e^T + e\tilde{v}_B^T,$$

where $\tilde{d}_B$ and $\tilde{v}_B$ are approximate reformulation vectors computed numerically by implementing (4.19). Instead of the exact B with eigenvalue decomposition $B = Q\Lambda Q^T$, we only have the original input matrix $\mathring{B}$ as well as the approximates $\tilde{d}_B$, $\tilde{v}_B$, $\tilde{\Lambda}$, and $\tilde{Q}$. Since $\tilde{Q}$ is a floating point matrix, we cannot expect orthogonality. For reasons of simplicity, the following explanation is limited to problem (4.32) and it is assumed that $d_A = v_A = 0$. If one places great emphasis on computing rigorous lower and upper bounds for the original definition of this framework, it is necessary to compute rigorous inclusions for d_B and v_B as well as an approximate $\tilde{Q}$ which satisfies $\tilde{Q}^T_{\Omega^0_+}\tilde{Q}_{\Omega^0_-} \equiv 0$. In practice, the validation of these presuppositions is of minor significance; we therefore skip the discussion of the corresponding details. We just suppose that $\tilde{d}_B, \tilde{v}_B$ describe the applied QAP reformulation and that $\big(\tilde{Q}_{\Omega^0_+}\tilde{\Lambda}_{\Omega^0_+}\tilde{Q}^T_{\Omega^0_+}, \tilde{Q}_{\Omega^0_-}\tilde{\Lambda}_{\Omega^0_-}\tilde{Q}^T_{\Omega^0_-}\big)$ states the used positive semidefinite matrix splitting of $\tilde{Q}\tilde{\Lambda}\tilde{Q}^T$.

The general idea is to assume that the approximates $\tilde{Q}$ and $\tilde{\Lambda}$ define the actual parameter matrix B. Rigorous bounds for the original problem are obtained by incorporating the quadratic residual term $\langle A, X(\mathring{B} - \tilde{Q}\tilde{\Lambda}\tilde{Q}^T)X^T\rangle$ into the linear term $\langle C, X\rangle$ of the objective function. Besides the easily transformable sum-matrix part introduced by the applied QAP reformulation, this concerns in particular the handling of the remainder

$$R = \mathring{B} + \operatorname{diag}^*(\tilde{d}_B) + \tilde{v}_B e^T + e\tilde{v}_B^T - \tilde{Q}\tilde{\Lambda}\tilde{Q}^T. \tag{7.23}$$

The open question is how to deal with the corresponding quadratic term $\langle A, XRX^T\rangle$ in the objective of the original QAP.

Usually, we will not be able to increase the computational accuracy to the point that R vanishes. On the other hand, it is possible to utilize algorithms for the computation of accurate dot products to obtain very tight inclusions for the actual remainder term given

on the right-hand side of (7.23). Efficient algorithms for these accurate computations are explained in [61, 78, 79, 118] and the references therein. The (interval) inclusion of (7.23) may be used to compute rigorous norm bounds for R. Together with appropriate estimates, such as

$$\forall X \in \mathcal{D}^n: \quad |\langle A, XRX^T\rangle| \leqslant \min\{|||A|||_{\mathrm{F}}|||R|||_{\mathrm{F}},\ |||A|||_{\mathrm{T}}|||R|||_2\},$$

we may obtain efficient bounds for the considered residual term $\langle A, XRX^T\rangle$. For tighter inclusions of this residual term, it is expedient to apply one of the discussed bounding techniques. The author suggests to use the Gilmore-Lawler bound procedure [37, 60] in order to obtain matrices L_R and U_R that satisfy

$$\forall X \in \Pi^n: \quad \langle L_R, X\rangle \leqslant \langle A, XRX^T\rangle \leqslant \langle U_R, X\rangle.$$

The final preparation step is the construction of interval matrices $\boldsymbol{B}_+$, $\boldsymbol{B}_-$, and $\boldsymbol{C}$ satisfying

$$\boldsymbol{B}_+ \ni \tilde{Q}_{\Omega_+^0}\tilde{\Lambda}_{\Omega_+^0}\tilde{Q}_{\Omega_+^0}^T, \quad \boldsymbol{B}_- \ni \tilde{Q}_{\Omega_-^0}\tilde{\Lambda}_{\Omega_-^0}\tilde{Q}_{\Omega_-^0}^T, \quad \text{and} \quad \boldsymbol{C} \supseteq \mathring{C} - 2Ae\tilde{v}_B^T + [L_R, U_R].$$

The corresponding interval data $(A, \boldsymbol{B}_+ - \boldsymbol{B}_-, \boldsymbol{C})$ contains a nonempty set of QAP instances with the same optima as the original problem $(\mathring{A}, \mathring{B}, \mathring{C})$. With all these preparations done, we are finally in the position to present an interval SDP problem that is suitable for the post-processing process implemented in VSDP. The corresponding interval programming problem

$$\mathbf{P} := \left\{ \mathrm{P}(A, \breve{B}_+, \breve{B}_-, \breve{C}, \tilde{Q}, \tilde{\Lambda}) \,\middle|\, \breve{B}_+ \in \boldsymbol{B}_+,\ \breve{B}_- \in \boldsymbol{B}_-,\ \breve{C} \in \boldsymbol{C} \right\} \tag{7.24}$$

specifies a family of SDP problems $\mathrm{P}(A, \breve{B}_+, \breve{B}_-, \breve{C}, \tilde{Q}, \tilde{\Lambda})$ defined via

$$\begin{aligned}
\inf_{X\in\mathcal{D}^n,\ Y_+,Y_-\in\mathcal{S}^n} \quad & \langle A, Y_+ - Y_-\rangle + \langle \breve{C}, X\rangle && \text{(7.25a)}\\
\text{s.t.} \quad & \begin{bmatrix} \tilde{\Lambda}_{\Omega_\diamond^0}^{-1} & \tilde{Q}_{\Omega_\diamond^0}^T X^T \\ X\tilde{Q}_{\Omega_\diamond^0} & Y_\diamond \end{bmatrix} \in \mathcal{S}_+, \qquad \diamond \in \{+,-\}, && \text{(7.25b)}\\
& \operatorname{diag}(Y_\diamond) = X \operatorname{diag}(\breve{B}_\diamond), \qquad \diamond \in \{+,-\}, && \text{(7.25c)}\\
& Y_+ e = X\breve{B}_+ e, \quad Y_- e = X\breve{B}_- e. && \text{(7.25d)}
\end{aligned}$$

Table 7.2.: Rigorous bounds for ill-posed MS_1 instances

Problem	d_{rel}	t_l/t	d^*_{rel}	t^*_l/t
Esc32g	$9.37 \cdot 10^{-8}$	0.623	$1.45 \cdot 10^{-7}$	$3.63 \cdot 10^{-4}$
Kra32	$2.43 \cdot 10^{-7}$	0.009	$3.15 \cdot 10^{-7}$	$2.63 \cdot 10^{-4}$
LiPa40a	$7.38 \cdot 10^{-10}$	1.020	$7.80 \cdot 10^{-10}$	$2.64 \cdot 10^{-4}$
Nug30	$1.27 \cdot 10^{-8}$	4.034	$9.19 \cdot 10^{-8}$	$3.42 \cdot 10^{-4}$
Ste36a	$2.10 \cdot 10^{-6}$	0.015	$2.10 \cdot 10^{-6}$	$2.72 \cdot 10^{-4}$
Tai35a	$5.37 \cdot 10^{-8}$	1.030	$5.38 \cdot 10^{-8}$	$3.37 \cdot 10^{-4}$
Tho40	$1.41 \cdot 10^{-6}$	1.015	$1.41 \cdot 10^{-6}$	$2.92 \cdot 10^{-4}$

Obviously, the minimal face reduction has not yet been applied to problem (7.25). Since the respective interval problems contain ill-posed programming instances, they typically also enclose infeasible programs. For this reason, it is not possible to compute upper bounds for these instances, and VSDP naturally fails to do so. On the other hand, VSDP is able to compute rigorous tight lower bounds. In Table 7.2, we present these bounds for selected instances of the interval problem (7.24). The quality of the computed lower bounds is measured in form of relative duality gaps defined as in (7.1). Due to the absence of rigorous upper bounds, d_{rel} and d^*_{rel} describe relative gaps between the computed rigorous lower bounds and the corresponding approximate upper bounds. We use d_{rel} to denote the gaps for rigorous lower bounds computed without any additional or non-default options in VSDP. The asterisk in d^*_{rel} refers to the computation of rigorous lower bounds with additional consideration of the boundedness of each optimization variable. This means that the respective VSDP function is called with the boundedness informations obtained a priori from the relaxation model. The relations between the computing time t for the approximation and the times t_l and t^*_l for the respective verified lower bound computations are displayed in the third and the fifth column of Table 7.2, respectively.

The rigorous bounds displayed in the fourth column of Table 7.2 are slightly worse than the bounds obtained via VSDP's post-processing. On the other hand, the accompanied computational costs are hardly perceptible. It is therefore beneficial to use the fast bounding procedure if possible. The numerical results given in Table 7.2 exemplify the

Table 7.3.: Rigorous bounds for well-posed MS_1 instances

Problem	d_{rel}	t_u/t	t_l/t	d^*_{rel}	t^*_l/t
Esc32g	$5.69 \cdot 10^{-4}$	1.097	1.238	$3.58 \cdot 10^{-5}$	$4.50 \cdot 10^{-4}$
Kra32	$4.89 \cdot 10^{-5}$	5.615	1.202	$2.12 \cdot 10^{-6}$	$3.41 \cdot 10^{-4}$
LiPa40a	$5.75 \cdot 10^{-5}$	2.036	1.042	$1.56 \cdot 10^{-8}$	$2.57 \cdot 10^{-4}$
Nug30	$1.35 \cdot 10^{-5}$	4.638	1.035	$4.45 \cdot 10^{-8}$	$3.92 \cdot 10^{-4}$
Ste36a	$4.02 \cdot 10^{-4}$	3.011	1.175	$4.74 \cdot 10^{-5}$	$3.29 \cdot 10^{-4}$
Tai35a	$1.45 \cdot 10^{-4}$	1.017	1.045	$1.08 \cdot 10^{-7}$	$3.92 \cdot 10^{-4}$
Tho40	$4.87 \cdot 10^{-4}$	2.007	0.992	$4.33 \cdot 10^{-6}$	$3.03 \cdot 10^{-4}$

possibility to compute rigorous lower bounds for the discussed relaxation frameworks with very low additional efforts. This is possible even without applying a minimal face reduction approach, and thereby without knowledge about the accompanied nullspace.

For the computation of verified upper bounds, it is necessary to apply the minimal face reduction described in Subsection 7.2.3. In the context of problem (7.25), it is sufficient to remove the last row and column from the block matrices in (7.25b). Considering the applicable cuts and their effect on the sparsity structure of the coefficient matrices, the elimination of the corresponding variables as demonstrated in Subsection 7.2.3 and 7.2.4 is not recommended since the transformation into implicit variables introduces additional intervals with larger radius, resulting in weaker inclusions.

In Table 7.3, we use a notation that is consistent with the one used in Table 7.2. The difference is that the presented relative gaps describe the quality of actual inclusions of the respective true optimal objective values because the upper bounds are computed rigorously as well. The computing time for obtaining verified upper bounds is denoted by t_u.

7.3. Implementation details

The inspection of implementation issues shall be completed by some final remarks on the exploitation of intrinsic problem structures. This comprises not only general properties

of the respective relaxation programs, such as the type and number of variables or the distribution of equality versus inequality constraints, but also the presence of low-rank and/or sparsity structures in the corresponding coefficient matrices.

7.3.1. Formulation in dual or primal standard form

A wide range of LP solvers, including IBM ILOG CPLEX [49], GLPK [38], Gurobi [43], and lp_solve [4], accept as input LP problems of the following form:

$$\begin{aligned} \inf_{x \in \mathbb{R}^n} \quad & \langle c, x \rangle \\ \text{s. t.} \quad & Ax \blacktriangleright b, \\ & l \leqslant x \leqslant u, \end{aligned} \tag{7.26}$$

where $A \in \mathcal{M}^{m,n}$, $b \in \mathbb{R}^m$, $\blacktriangleright \in \{=, \geqslant, \leqslant\}^m$, and $l, u \in \{\mathbb{R} \cup \{\pm\infty\}\}^n$. Indeed, most of these solvers support even more general formulations of the respective LP instance. Obviously, problem (7.26) covers both the primal standard form

$$\begin{aligned} \inf_{x \in \mathbb{R}^n} \quad & \langle c, x \rangle \\ \text{s. t.} \quad & Ax = b, \\ & x \geqslant 0, \end{aligned}$$

as well as its dual

$$\begin{aligned} \sup_{y \in \mathbb{R}^m} \quad & \langle b, y \rangle \\ \text{s. t.} \quad & A^T y \leqslant c. \end{aligned}$$

The greater flexibility of formulation (7.26) allows a good exploitation of the individual problem structure. For a wide range of LP instances, this provides the possibility to construct quite efficient implementations, more efficient than it would have been possible in one of the two standard forms. This *improved efficiency* comprises the simplicity of bringing a given LP model into formulation (7.26), a reduced memory usage, and usually also a faster solution procedure.

Consider, for instance, the minimization problem

$$\begin{aligned} \inf_{x\in\mathbb{R}^2} \quad & 2x_1 - x_2 \\ \text{s. t.} \quad & x_1 + x_2 = -2, \\ & 0 \leqslant x_2 \leqslant 3. \end{aligned}$$

Though this problem is trivial, it suffices for demonstrating the benefit of formulation (7.26). The transformation of this problem into primal standard form requires four variables and two equality conditions. For a formulation in dual standard form, only two variables but four inequality constraints are needed. Formulation (7.26) allows the most efficient implementation, since only two variables (with lower and upper bounds) and one equality condition are required. If a primal-dual approach is used to solve the LP problem, it is also important to consider the number of variables and constraints in the corresponding dual problem. Even in this context, the general advantage of formulation (7.26) can be shown.

Unfortunately, solvers that support the cone of semidefinite matrices are usually requiring rather restricted formulations for their input format. In contrast to pure SDP solvers - which typically allow the exploitation of block-diagonal structures but are otherwise limited to semidefiniteness conditions -, conic solvers support the optimization over *direct products* of different convex cones, such as semidefinite cones, second-order cones and nonnegative orthants. Solvers like CSDP [9] or SDPA [112] fall into the first category; SeDuMi [101], SDPT3 [102, 103], MOSEK [75], and SCS [77], for example, belong to the class of conic solvers. All these solvers require that the considered problem is given in some format which is an extension to either (PSDP) or (DSDP), but does not cover both formulations. This is typically caused by the underlying algorithms that require one of these standard formulations.

For reasons of simplicity, let us assume that LP variables can be treated efficiently in form of 1×1 semidefinite matrices and that second-order cone variables are not present. In this context, it is sufficient to consider the formulations (PSDP) and (DSDP) with an underlying block-diagonal structure. The input format for the applied solver shall be designed for problems given either in the form (PSDP) or (DSDP). Since these formulations are dual to each other, it is possible to choose the form that is more suitable for the respective relaxation.

In order to demonstrate the benefit in considering both standard forms, we will evaluate the corresponding formulations for a relaxation based on the matrix lifting approach:

$$\inf_{X\in\mathcal{N}^n,\ Y,Z\in\mathcal{S}^n} \quad \langle A, Y\rangle + \langle C, X\rangle \tag{7.27a}$$

$$\text{s. t.} \quad \begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \geq 0, \tag{7.27b}$$

$$\operatorname{diag}(Y) = X\operatorname{diag}(B), \quad \operatorname{diag}(Z) = X\operatorname{diag}(B^2), \tag{7.27c}$$

$$\langle E, Z\rangle = \langle E, B^2\rangle, \tag{7.27d}$$

$$Xe = X^Te = e, \tag{7.27e}$$

$$Xv_le^T + ew_l^TX^T \leqslant_{\text{off}} Y \leqslant_{\text{off}} Xv_ue^T + ew_u^TX^T. \tag{7.27f}$$

Problem (7.27) is described via $2n^2 + n$ variables (of different type), a semidefiniteness condition in $\mathcal{S}^{3n}$, $2n^2 - 2n$ linear inequalities and $4n$ equality constraints (not counting the single redundant equality in (7.27e)). Although there is still room for improvements, this formulation states already a quite efficient realization of the corresponding relaxation program.

Let us now consider a reformulation in primal standard form. In this formulation, inequalities are implemented by introducing adequate conic variables. The links to already introduced variables are established via additional equality constraints. To formulate an equivalent problem to (7.27) in primal standard form, we introduce a matrix variable $H \in \mathcal{S}_+^{3n}$ that is used to realize the semidefiniteness condition in (7.27b). Moreover, we introduce two nonnegative matrix variables $\underline{Y}, \overline{Y} \in \mathcal{N}^n$ to implement the linear inequality constraints in (7.27f). For a better link to the variables X, Y, and Z in problem (7.27), we handle H as a block matrix arranged as

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix},$$

where all matrix blocks $\{H_{ij}\}$ are of dimension $n \times n$. By taking all these things together, we obtain the following formulation:

$$\inf_{H \in \mathcal{S}_+^{3n},\ X, \underline{Y}, \overline{Y} \in \mathcal{N}^n} \quad \langle A, H_{32} \rangle + \langle C, X \rangle \tag{7.28a}$$

$$\text{s. t.} \quad H_{11} = H_{22} = I, \quad H_{21} = X, \quad H_{31} = XB, \quad H_{32} = H_{32}^T, \tag{7.28b}$$

$$\operatorname{diag}(H_{31}) = X \operatorname{diag}(B), \quad \operatorname{diag}(H_{33}) = X \operatorname{diag}(B^2), \tag{7.28c}$$

$$\langle E, H_{33} \rangle = \langle E, B^2 \rangle, \tag{7.28d}$$

$$Xe = X^T e = e, \tag{7.28e}$$

$$\begin{aligned} \underline{Y} &=_{\text{off}} H_{32} - X v_l e^T - e w_l^T X^T, \\ \overline{Y} &=_{\text{off}} X v_u e^T + e w_u^T X^T - H_{32}. \end{aligned} \tag{7.28f}$$

The corresponding equality constraints are in one-to-one correspondence to the respective conditions in problem (7.27). The semidefiniteness condition is now incorporated into the variable H. However, the interrelations between its components and the link to X has to be reestablished via the equality conditions in (7.28b). We even need constraints to reestablish the symmetry of the matrix block H_{32} which corresponds to the variable Y. Taking the symmetry of H into account, (7.28b) describes $\frac{7}{2}n^2 + \frac{1}{2}n$ non-redundant equality constraints. In total, problem (7.28) requires $\frac{15}{2}n^2 - \frac{1}{2}n$ variables (not counting the diagonal elements of $\underline{Y}$, $\overline{Y}$) and $\frac{11}{2}n^2 + \frac{5}{2}n$ equality constraints.

The construction of an equivalent reformulation that complies with the dual standard form is more straightforward. Equality constraints can be implemented via two oppositely directed inequalities.[3] Conic properties of the respective variables have to be realized via

[3] A way to circumvent the accompanied numerical problems will be described in the next subsection.

additional conditions.[4] The respective formulation is given by

$$\inf_{X\in\mathcal{M}^n,\ Y,Z\in\mathcal{S}^n} \quad \langle A, Y\rangle + \langle C, X\rangle \tag{7.29a}$$

$$\text{s. t.} \quad \begin{bmatrix} I & X^T & BX^T \\ X & I & Y \\ XB & Y & Z \end{bmatrix} \geq 0, \tag{7.29b}$$

$$\begin{aligned} &X\,\mathrm{diag}(B) \leqslant \mathrm{diag}(Y) \leqslant X\,\mathrm{diag}(B), \\ &X\,\mathrm{diag}(B^2) \leqslant \mathrm{diag}(Z) \leqslant X\,\mathrm{diag}(B^2), \end{aligned} \tag{7.29c}$$

$$\langle E, B^2\rangle \leqslant \langle E, Z\rangle \leqslant \langle E, B^2\rangle, \tag{7.29d}$$

$$X \geqslant 0, \quad e \leqslant Xe \leqslant e \leqslant X^T e \leqslant e, \tag{7.29e}$$

$$Xv_l e^T + ew_l^T X^T \leqslant_{\text{off}} Y \leqslant_{\text{off}} Xv_u e^T + ew_u^T X^T. \tag{7.29f}$$

Formulation (7.29) is described via $2n^2 + n$ free variables, a semidefiniteness condition in $\mathcal{S}^{3n}$, and $3n^2 + 6n$ linear inequality constraints. For a meaningful comparison of problem (7.28) and (7.29), it is important to consider the same form, that is either the primal of (7.28) with the dual of (7.29) or the other way around. Let us consider the first case. The dual problem of (7.29) involves $\frac{15}{2}(n^2 + n)$ variables and $2n^2 + n$ equality constraints. In comparison to the primal problem of (7.28) which is realized via $\frac{15}{2}n^2 + \frac{1}{2}n$ variables and $\frac{11}{2}n^2 + \frac{5}{2}n$ equality constraints, the number of variables is slightly higher whereas the number of equalities is significantly smaller. In the view of this situation, formulation (7.29) is definitely the preferable choice for the actual implementation of problem (7.27).

Let us conclude the discussion of possible formulations of the presented SDP frameworks. Due to restrictions in the input format of the respective solvers, it is usually necessary to use a formulation that conforms to either the standard primal or the standard dual form. As seen exemplary for problem (7.27), the choice of this form can have a significant influence on the number of accompanied equality constraints in the primal standard form (= number of variables in its dual). For relaxation frameworks based on matrix lifting, such as problem (7.27), the dual form is more beneficial. On the other hand, the "eigenspace" SDP relaxation involves significantly fewer equality constraints when implemented in the standard primal form. It is therefore important to ask, which formulation is more suitable for a given SDP problem. The choice depends on the structure of the corresponding conic inequalities. If the number of variables (counting

[4] They are transfered into dual slack variables.

each component) that are necessary to describe these inequalities is small compared to the number of inequality constraints (again counting every component), then a formulation that complies with the dual standard form is preferable. On the other hand, if there are relatively few interrelations within and/or between the respective inequality conditions, a formulation in primal standard form often results in a more efficient implementation. In this context, it is apparent that we implement the vector lifting based relaxation programs as well as the "eigenspace" SDP relaxations in compliance with the primal standard form. Conversely, the frameworks based on the matrix splitting or the matrix lifting technique are implemented in respect of the dual standard form. At this point, we should mention that the compared frameworks in Table 7.1 have all been formulated in compliance with the same standard form, there the dual one. The intentional choice of the inappropriate primal standard formulation would have given even worse results for the original frameworks.

7.3.2. Further implementation details and low-rank coefficient structures

It is worth mentioning that the reformulation procedure discussed in the previous subsection as well as its dualization can be automated, see [40, 41] and [64, 65]. Nevertheless, a manual reformulation of the respective relaxation framework may still be advantageous because it gives a better control about implementation details that are relevant for the running times of the corresponding solving processes.

As an example, consider problem (7.29) from the last subsection. If the input format of the used solver covers the primal standard form but in addition also supports free variables, then the corresponding dual also allows a *direct* implementation of equality constraints. This results in realizations without the necessity of tricks such as the use of oppositely directed inequalities. The default solvers used by CVX [41], including SeDuMi [101] and SDPT3 [102, 103], support free variables in their primal form. The CVX package makes use of this capability and avoids the implementation of equality constraints via inequalities. Nevertheless, it is important to notice that - though the mentioned solvers support free variables - they do not handle them very well. Internally, these solvers do similar transformations from free variables to conic ones. For this reason, we try to reduce the number of explicit equalities.

The diagonal elements of the matrix variables Y and Z are used only once in (7.29b). It is therefore beneficial to avoid the explicit use of these variables. The corresponding constraints in (7.29c) can be removed if we replace (7.29d) with $\langle E_{\text{off}}, Z\rangle \leqslant \langle E_{\text{off}}, B^2\rangle$ and substitute

$$\begin{bmatrix} I & X^T & BX^T \\ X & I & Y_{\text{off}} + \operatorname{diag}^*(X \operatorname{diag}(B)) \\ XB & Y_{\text{off}} + \operatorname{diag}^*(X \operatorname{diag}(B)) & Z_{\text{off}} + \operatorname{diag}^*(X \operatorname{diag}(B^2)) \end{bmatrix} \succeq 0$$

for (7.29b). As a result of these substitutions, the diagonal components of Y and Z have no coefficients in any of the constraints or the objective function. The corresponding variables may therefore be removed.

Moreover, by the same procedure as the one we have already used for the minimal face reduction of the respective problems, we can eliminate the constraints which implement the condition $X \in \mathcal{E}^n$. In the style of (7.21e), define the transformation $\mathbf{X}\colon \mathcal{M}^{n-1} \to \mathcal{M}^n$ as follows:

$$\mathbf{X}(\tilde{X}) = \begin{bmatrix} \tilde{X} & (I - \tilde{X})e \\ e^T(I - \tilde{X}) & e^T(\tilde{X} - I)e \end{bmatrix}. \tag{7.30}$$

It is then possible to replace the variable $X \in \mathcal{M}^n$ in problem (7.29) with $\tilde{X} \in \mathcal{M}^{n-1}$ and substitute $\mathbf{X}(\tilde{X})$ for every other occurrence of X. For the compliance with the conditions in (7.29e), it is sufficient to require $\mathbf{X}(\tilde{X}) \succcurlyeq 0$.

Very similar reduction strategies are applicable to the respective formulations of other relaxation frameworks. For the design of the actual implementations, we combine these strategies with the facial reduction procedure discussed in the previous section.

Before we finally get to the numerical results and the conclusion of this work, let us complete the inspection of implementation details with two further examples of possible improvements. We begin with a beneficial utilization of intermediate variables. In the first part of this subsection, we reviewed a simple strategy to reduce the number of equality constraints via elimination of variables. This approach works well for formulation (7.29) because the removed variables are involved in very few other constraints. In many other cases, the elimination of variables would be counterproductive. The reason for this is that the elimination of variables typically destroys the sparsity structure of the corresponding coefficient matrices. By introducing additional variables for intermediate calculations,

we aim for the opposite effect: more variables in exchange for a more beneficial sparsity structure of the coefficient matrices.

A good example for a beneficial use of additional variables that keep results of intermediate computations has already been emphasized in Subsection 6.1.2. Without further modifications, the upper bound constraints in (6.25) involve $n^4 + 2n^2$ coefficients (n^2+2 coefficients for each inequality). By the introduction of n^2 additional variables, this number can be reduced to $2n^3 + 2n^2$. It usually makes a huge difference for the applied solving method whether an SDP problem involves $\mathcal{O}(n^4)$ or $\mathcal{O}(n^3)$ nonzero coefficients. Beyond such extreme cases, the introduction of additional variables for intermediate results can also be sensible for less expensive constraints such as (7.29f). In the actual implementation, we introduce $4n$ additional variables in form of four n-component vectors $v_{lx}, w_{lx}, v_{ux}, w_{ux} \in \mathbb{R}^n$ and substitute

$$Xv_l \leqslant v_{lx}, \quad Xw_l \leqslant w_{lx}, \quad v_{ux} \leqslant Xv_u, \quad w_{ux} \leqslant Xw_u, \tag{7.31a}$$

$$v_{lx}e^T + ew_{lx}^T \leqslant_{\text{off}} Y \leqslant_{\text{off}} v_{ux}e^T + ew_{ux}^T \tag{7.31b}$$

for (7.29f). Since problem (7.29) is formulated with respect to the dual standard form, it is recommended to avoid the introduction of unnecessary equality constraints. The utilization of the inequalities in (7.31a) is more practical than the incorporation of their equality counterparts and the cutting effect on the matrix variable Y is the same.

In certain cases, it can be beneficial to use the variables which are introduced to keep the results of intermediate computations as substitutes for original variables. Consider, for instance, the relaxation based on the inverse interrelated matrix splitting approach. The block matrix for the semidefiniteness condition in (5.29b) contains the two matrix blocks $XB_{\triangle}^{\frac{1}{2}}$ and $XB_{\triangledown}^{\frac{1}{2}}$. Actually, after the multiplication with the block-diagonal matrix D_Y, the corresponding blocks are $XB_{\triangle}^{\frac{1}{2}}Q_{\Omega^\varepsilon}$ and $XB_{\triangledown}^{\frac{1}{2}}Q_{\Omega^\varepsilon}$, respectively. For reasons of simplicity, assume that B is of full rank and that the approximation threshold ε is set to zero. Under these assumptions, we have

$$XB_{\diamond}^{\frac{1}{2}}Q_{\Omega^\varepsilon} = XB_{\diamond}^{\frac{1}{2}}Q = XQ\Lambda_{\diamond}^{\frac{1}{2}} \qquad \text{for} \quad \diamond \in \{\triangle, \triangledown\},$$

where $B_{\triangle} = Q\Lambda_{\triangle}Q^T$ and $B_{\triangledown} = Q\Lambda_{\triangledown}Q^T$ denote the spectral decompositions of the corresponding matrices. By introducing a new matrix variable $X_Q \in \mathcal{M}^n$ for the term XQ and substituting X_QQ^T for the original variable X, we reformulate problem (5.29)

and obtain

$$\inf_{X_Q\in\mathcal{M}^n,\ G,Y,Y_\triangle,Y_\triangledown\in\mathcal{S}^n} \quad \langle A, Y\rangle + \langle CQ, X_Q\rangle \tag{7.32a}$$

$$\text{s. t.} \quad \begin{bmatrix} I & \Lambda_\triangle^{\frac{1}{2}} X_Q^T & \Lambda_\triangledown^{\frac{1}{2}} X_Q^T \\ X_Q\Lambda_\triangle^{\frac{1}{2}} & Y_\triangle & G \\ X_Q\Lambda_\triangledown^{\frac{1}{2}} & G & Y_\triangledown \end{bmatrix} \succeq 0, \tag{7.32b}$$

$$\begin{bmatrix} I_{\mathfrak{V}_G^1}^T \left(D_\tau - \Lambda_\triangle^{\frac{1}{2}}\Lambda_\triangledown^{\frac{1}{2}}\right)^\dagger I_{\mathfrak{V}_G^1} & I_{\mathfrak{V}_G^1}^T X_Q^T \\ X_Q I_{\mathfrak{V}_G^1} & D_\tau - G \end{bmatrix} \succeq 0, \tag{7.32c}$$

$$\begin{aligned} &\operatorname{diag}(Y_\diamond) = X_Q\big(Q^T \operatorname{diag}(B_\diamond)\big) \qquad \text{for} \quad \diamond \in \{\triangle, \triangledown\}, \\ &\operatorname{diag}(G) = X_Q\big(Q^T \operatorname{diag}(B_\triangle^{\frac{1}{2}} B_\triangledown^{\frac{1}{2}})\big), \end{aligned} \tag{7.32d}$$

$$\langle Y_\triangle, E\rangle = \langle B_\triangle, E\rangle, \quad \langle Y_\triangledown, E\rangle = \langle B_\triangledown, E\rangle, \tag{7.32e}$$

$$Y = Y_\triangle - Y_\triangledown, \tag{7.32f}$$

$$X_Q Q^T \geqslant 0, \quad X_Q(Q^T e) = e, \quad X_Q^T e = Q^T e, \tag{7.32g}$$

where the index set $\mathfrak{V}_G^1 := \{i \mid |\lambda_i(B_\triangle^{\frac{1}{2}} B_\triangledown^{\frac{1}{2}})| < \|\!|B_\triangle^{\frac{1}{2}} B_\triangledown^{\frac{1}{2}}|\!\|_2\}$ and the diagonal matrix $D_\tau := \tau(B)I$ comply with their definitions in Subsection 5.1.3.

The orthogonal matrix Q typically contains very few zero components. Under the assumption that Q is full or at least almost full and that the dimension is not too small ($n > 16$), the implementation of problem (7.3.2) requires only about half as many nonzero coefficients as the realization of the original formulation (5.29). Moreover, numerical tests have shown that the described modification accelerates the solving procedure noticeably.

Our final remark concerns the capability of some conic solvers to exploit low-rank structures in coefficient matrices. The tools SDPLR [13, 14], SDPT3 [102, 103], LMI LAB [36], as well as modified versions of CSDP [50] cover just a small selection of SDP solvers which support low-rank input data. In the view of the discussed relaxation frameworks, making use of this possibility is particularly beneficial for the implementation of *EVB* related inequalities.

To be more precise, here we consider the constraints in (6.17). A rewrite of these constraints for compliance with the dual standard form gives

$$\left\langle Y, \sum_{i=2}^{l} p_i p_i^T \right\rangle \geqslant \sum_{i=2}^{l} \lambda_i \qquad \text{for} \quad l \in \{2, \ldots, n-1\}. \tag{7.33}$$

Apparently, the rank of the corresponding coefficient matrices depends on the summation limit l. However, with a very similar trick as before - which is referring to the introduction of variables for intermediate calculations -, it is possible to reformulate (7.33) in such a way that only rank-one coefficient matrices are present. For this purpose, one needs to incorporate $n-2$ additional variables $r \in \mathbb{R}^{n-2}$ and replace (7.33) with

$$\begin{aligned} & r \geqslant 0, \quad r_1 \leqslant \left\langle Y, p_2 p_2^T \right\rangle - \lambda_2, \\ & r_{l-1} - r_{l-2} \leqslant \left\langle Y, p_l p_l^T \right\rangle - \lambda_l \qquad \text{for} \quad l \in \{2, \ldots, n-2\}. \end{aligned} \tag{7.34}$$

If the low-rank structure in (7.34) is also accounted in the process of generating the input data for the respective solver, then the SDP problem is handled more efficiently.

The counterpart formulation to (7.34) for compliance with the primal standard form can be realized even more straightforwardly. On top of this, the respective formulation is more efficient since it does not require the introduction of additional variables. The modification for the actual used EVB constraints given in (6.45) follows the same approach.

7.4. Numerical results

After having specified the final model of constraints for the respective level-2 relaxations in Section 6.2 and after having given the necessary explanations for their actual implementation in the last two sections, it is time to discuss the practical applicability of these relaxations on the basis of numerical tests. Since this section serves the purpose of evaluating the quality and applicability of the finalized versions of our SDP frameworks, the numerical results are significantly more extensive than in the previous chapters. Here we give bounds for all symmetric problem instances from the QAP library [18] up to dimension $n = 99$.

As already mentioned in Subsection 7.3.1, the implementation of VL_2, ES_2, and ESC_2 is done in compliance with the primal standard form (PSDP). The low-dimensional relaxations ML_2, MLX_2, MS_2, and $IIMS_2$ are implemented with regard to the dual standard form (DSDP). The practical realization of the minimal face reduction described in Subsection 7.2.4 is applied to all these frameworks. For the implementation of the SDP relaxations which are formulated in dual standard form, also the strategies described in Subsection 7.3.2 are exploited. This comprises the ideas to reduce the number of

equality constraints as well as the substitution of the matrix variable X_Q for XQ. The reformulation of the additional LP constraints via (7.31) is again used on all relaxations. However, the author refrained from using reformulation techniques whose exploitation requires specific capabilities of the applied solver. This includes the substitute (7.34) for the conditions in (7.33).

The computational costs for solving ML_2 are usually less than the ones for computing the solution to the respective MLX_2 instances; indeed the measured computing times were very similar to the ones for $IIMS_2$. However, ML_2 clearly falls behind its successor MLX_2 when it comes to bounding quality. The framework ML_2 is therefore omitted from the subsequent evaluation.

The numerical results are presented following the style in [83]. As before, the bounds are given in form of relative gaps

$$R_{\text{gap}} := 1 - \frac{\text{Lower bound computed via relaxation}}{\text{Best known upper bound or optimal value}}.$$

Under the 'CPU' columns, the corresponding computing times are listed in seconds. The labels in the column 'Problem' consists of three or four letters indicating the names of the authors or contributors of the respective QAP instance, together with a number that gives their dimension. If the authors provided multiple problem instances for the same dimension, the respective instance is indicated by another letter at the end of the name. For more information on the naming scheme and the individual applications, see [18].

Our testing software sdprQAP is written in MATLAB and utilizes the modeling language YALMIP [64] in many parts of the code. For the computation of the bounds presented in Table 7.4 and 7.5, we used the solver SDPT3 [102] that implements an infeasible primal-dual path-following interior-point algorithm. The interpretation of the code of sdprQAP is done by Octave [29].

Table 7.4.: Numerical results for instances with size $n < 30$

	VL_2		ES_2		$ESC_2{}^{**}$	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Chr12a	4.85	193	10.16	110	10.64	103
Chr12b	Fail	–	16.20	113	16.34	84

continued ...

	VL_2		ES_2		ESC_2 **	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Chr12c	6.78	118	10.68	161	11.14	99
Chr15a	11.40	253	19.71	231	21.53	152
Chr15b	18.93	287	30.71	196	33.95	145
Chr15c	0.25	288	5.89	228	8.80	190
Chr18a	8.44*	380	14.18	295	15.95	210
Chr18b	0.00	297	0.00	309	0.00	168
Chr20a	1.58	632	1.64	467	1.64	351
Chr20b	2.07	582	2.53	433	2.72	271
Chr20c	11.66*	686	18.48	401	18.95	221
Chr22a	1.01	1049	2.73	479	3.03	319
Chr22b	1.07	1075	2.35	507	2.73	338
Chr25a	6.49	1811	13.73	549	15.89	444
Els19	2.00*	977	Fail	–	Fail	–
Esc16a	14.71	251	17.14	179	17.14	55
Esc16b	2.74	204	2.74	226	2.74	48
Esc16c	12.50	265	16.10	255	16.10	102
Esc16d	34.37	222	80.23	197	80.23	59
Esc16e	33.12	232	37.11	206	37.11	65
Esc16f	0.00	0	0.00	0	0.00	0
Esc16g	30.56	445	30.56	174	30.56	53
Esc16h	2.04	218	2.07	200	2.07	92
Esc16i	18.78	229	100.00	164	100.00	121
Esc16j	11.13	208	89.21	174	89.21	60
Had12	0.46	117	1.30	111	2.03	73
Had14	0.47	193	1.26	171	1.50	138

continued ...

	VL_2		ES_2		${ESC_2}^{**}$	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Had16	1.02	275	1.97	223	2.31	165
Had18	1.16	379	2.06	299	2.42	205
Had20	0.87	641	1.69	350	2.15	219
LiPa20a	0.12	557	0.38	373	0.48	224
LiPa20b	0.00	609	0.00	441	0.02	355
Nug12	7.06	136	8.18	105	11.94	110
Nug14	4.56	234	6.73	134	9.07	127
Nug15	5.58	298	6.73	177	8.13	139
Nug16a	4.75	372	5.80	210	7.73	140
Nug16b	7.45	371	8.55	200	13.27	131
Nug17	6.03	487	8.34	218	10.21	152
Nug18	6.43	553	7.90	229	9.92	180
Nug20	6.88	877	7.76	304	8.94	219
Nug21	6.51	1181	7.67	318	9.38	247
Nug22	5.17	1530	6.88	354	8.86	321
Nug24	6.77	2360	7.54	423	7.86	295
Nug25	6.92	2875	7.68	439	7.77	386
Nug27	5.13	3128	6.50	511	7.12	487
Nug28	6.52	3973	7.45	574	8.80	570
Rou12	5.90	157	8.79	109	10.89	111
Rou15	8.58	298	11.21	152	13.78	164
Rou20	11.34	1032	13.89	289	16.29	276
Scr12	5.54	148	7.04	118	10.07	119
Scr15	10.10	319	18.96	181	20.13	164
Scr20	12.80	1007	14.38	376	16.92	295

continued ...

	VL_2		ES_2		ESC_2 **	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Tai12a	3.65	164	6.79	115	9.00	117
Tai12b	3.75	257	Fail	–	Fail	–
Tai15a	9.73	302	12.29	162	14.11	175
Tai15b	Fail	–	0.47	113	0.55	124
Tai17a	10.11	374	12.52	253	14.64	196
Tai20a	11.93	716	14.65	376	16.54	273
Tai20b	Fail	–	20.12	380	Fail	–
Tai25a	13.28	3016	15.39	536	17.34	453

* solved with numerical problems, accuracy below 5 decimal digits

** approximation tolerance $\varepsilon = 0.1$

Table 7.5.: Numerical results for instances with size $n < 100$

	MLX_2		MS_2		$IIMS_2$	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Chr12a	10.76	3	11.41	2	11.12	8
Chr12b	15.15	3	15.40	2	15.34	8
Chr12c	11.37	3	11.83	2	11.16	9
Chr15a	20.67	6	22.21	3	21.55	4
Chr15b	31.32	5	32.56	3	32.10	4
Chr15c	7.51	5	8.79	3	7.85	4
Chr18a	15.31	9	16.32	4	15.63	6
Chr18b	0.00	7	0.00	4	0.00	5
Chr20a	1.64	14	1.64	7	1.64	10
Chr20b	2.58	14	2.74	7	2.66	10

continued ...

	MLX_2		MS_2		$IIMS_2$	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Chr20c	19.89	13	19.08	7	18.90	9
Chr22a	3.04	23	3.28	9	3.09	13
Chr22b	2.58	23	2.84	10	2.66	13
Chr25a	Fail	–	Fail	–	Fail	–
Els19	Fail	–	13.09	7	Fail	–
Esc16a	24.02	4	24.02	3	19.82	4
Esc16b	4.83	4	4.83	3	3.96	5
Esc16c	23.70	4	23.88	3	21.12	5
Esc16d	75.00	4	75.00	3	75.00	4
Esc16e	49.84	5	49.84	3	45.80	5
Esc16f	0.00	0	0.00	0	0.0	0
Esc16g	40.55	5	35.41	3	32.84	6
Esc16h	3.64	4	2.07	3	2.82	4
Esc16i	100.00	4	100.00	2	100.00	3
Esc16j	75.01	4	75.02	3	75.01	4
Esc32e	100.00	60	100.00	17	100.00	36
Esc32g	100.00	53	100.00	17	100.00	39
Had12	1.51	2	1.78	2	1.71	9
Had14	1.76	3	2.26	3	2.13	3
Had16	2.30	4	3.01	3	2.90	4
Had18	2.77	7	2.88	4	2.78	6
Had20	2.46	9	2.33	6	2.31	8
Kra30a	13.40	54	18.01	17	16.56	45
Kra30b	14.48	57	18.93	17	17.58	43
Kra32	15.34	79	20.49	22	19.06	54

continued ...

	MLX_2		MS_2		$IIMS_2$	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
LiPa20a	0.33	12	0.39	6	0.36	8
LiPa20b	0.00	17	0.01	7	0.00	11
LiPa30a	0.42	71	0.44	31	0.43	40
LiPa30b	0.00	92	0.01	27	0.00	58
LiPa40a	0.30	270	0.30	82	0.30	142
LiPa40b	0.00	357	0.03	83	0.01	170
LiPa50a	0.24	795	0.24	516	0.24	975
LiPa50b	0.00	1622	0.05	504	0.03	950
LiPa60a	0.22	2832	0.22	979	0.22	1498
LiPa60b	0.01*	2385	0.03*	764	0.03*	1101
LiPa70a	0.16	4046	0.17	1144	0.17	1974
LiPa70b	0.01*	4721	0.05	1123	0.03*	2200
LiPa80a	0.12	7890	0.12	2039	0.12	3389
LiPa80b	0.02	9149	0.06	1947	0.05	4848
LiPa90a	0.10	19 356	0.10	4721	0.10	8209
LiPa90b	0.03	17 545	0.06	3696	0.05	6660
Nug12	9.59	2	11.83	2	11.59	2
Nug14	8.45	4	7.61	3	7.54	4
Nug15	8.52	4	7.78	3	7.64	4
Nug16a	6.45	5	6.93	4	6.59	4
Nug16b	9.36	4	9.14	4	9.23	4
Nug17	Fail	–	9.35	4	8.93	5
Nug18	8.82	7	9.32	4	8.69	7
Nug20	8.00	11	8.76	4	8.32	8
Nug21	Fail	–	Fail	–	Fail	–

continued ...

	MLX_2		MS_2		$IIMS_2$	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Nug22	10.51	13	10.21	9	9.96	12
Nug24	7.94	22	8.82	7	8.16	16
Nug25	8.13	21	Fail	–	Fail	–
Nug27	Fail	–	Fail	–	Fail	–
Nug28	8.22	41	8.35	13	8.17*	27
Nug30	7.84	56	7.95	18	7.89	39
Rou12	10.05	2	10.90	2	10.40	9
Rou15	12.69	4	13.17	3	12.94	4
Rou20	15.01	12	16.04	6	15.51	9
Scr12	7.69	2	10.64	2	10.55	3
Scr15	15.01	5	14.92	3	14.99	3
Scr20	14.36	12	15.21	4	14.90	9
Sko42	7.41	246	7.21	78	7.09	157
Sko49	Fail	–	Fail	–	Fail	–
Sko56	6.91	970	6.53	230	6.47	642
Sko64	6.48	1358	6.08	424	5.97	1172
Sko72	6.07	3371	5.58	752	5.53	2037
Sko81	Fail	–	Fail	–	Fail	–
Sko90	5.90	11 459	5.20	2189	5.20	6354
Ste36a	18.11	143	20.61	41	18.37	96
Ste36b	15.85	115	20.45	49	16.32*	103
Ste36c	15.64	217	14.94	66	12.89	107
Tai12a	7.64	2	8.84	2	8.14	9
Tai12b	11.59	3	8.21	2	10.43	4
Tai15a	13.11	4	13.92	3	13.59	32

continued ...

	MLX_2		MS_2		$IIMS_2$	
Problem	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU	$R_{\text{gap}}(\%)$	CPU
Tai15b	0.62	6	0.68	4	0.73	5
Tai17a	Fail	–	14.59	3	14.08	5
Tai20a	15.59	12	16.33	6	16.01*	14
Tai20b	8.30	12	4.70	9	5.66	12
Tai25a	Fail	–	Fail	–	Fail	–
Tai25b	Fail	–	Fail	–	Fail	–
Tai30a	15.95*	74	16.74*	27	16.34*	40
Tai30b	18.05	60	10.76	34	12.50	51
Tai35a	Fail	–	Fail	–	Fail	–
Tai35b	22.89	105	13.58	64	15.76	96
Tai40a	19.72*	541	20.28*	87	19.99	165
Tai40b	Fail	–	10.06	103	11.32	493
Tai50a	21.22	1480	21.63*	394	21.48	507
Tai50b	17.21	742	12.72	245	13.84	450
Tai60a	22.28*	1895	22.76	544	22.55	1237
Tai60b	17.68	1833	10.13	564	11.62	1105
Tai64c	55.52	3146	2.41	476	2.40	932
Tai80a	23.08	9962	23.38	2190	23.27	4111
Tai80b	17.71	9820	11.20	2211	11.98	3797
Tho30	13.61	44	12.17	20	11.77	40
Tho40	12.48	208	12.45	77	12.17	134
Wil50	3.88	589	3.61	202	3.58	354

* solved with numerical problems, accuracy below 5 decimal digits

The results presented in Table 7.4 reveal a significant superiority of VL_2 compared to all other tested relaxations. Even without the integration of the nonnegativity condition

$\Upsilon \geqslant 0$, we obtain very strong bounds for the respective QAP instances. Unfortunately, for dimensions $n \geqslant 30$ the interior-point algorithm becomes too expensive for our hardware, which is why the corresponding problems had to be omitted from the numerical tests. The given bounds also reveal the limitations of the eigenvalue clustering approach. Even the moderate approximation threshold $\varepsilon = 0.1$ often leads to a noticeable weakening of ESC_2 compared to ES_2. The computing times of these two frameworks are quite similar. Indeed, only the instances with exploitable low-rank parameter matrices benefit noticeably from the eigenspace clustering approach.

Table 7.5 shows a different picture for the low-dimensional relaxation frameworks. The bounds are typically quite close to each other. Overall, MLX_2 is the best performing framework amongst the low-dimensional SDP relaxations. However, it is also more expensive than its competitors. It is difficult to determine the most efficient relaxation approach since the bounding quality depends too much on the respective problem classes.

The interior-point method becomes too expensive for SDP relaxations of higher dimensions. It is possible to use different solving procedures such as operator splitting or Newton-CG augmented Lagrangian methods, see [77] or [113, 117]. However, for the considered ill-conditioned relaxations, the corresponding solvers SCS and SDPNAL+ tend to produce inaccurate results. Using the regularization technique described in Subsection 7.2.2 together with the methods introduced in [53, 54], it is still possible to compute (verified) lower bounds for the optimal objective values to these problems. However, both approaches are sensitive to the condition measure of the problem. In combination with an inaccurate solving method, the obtained bounds can lie considerably below the actual optimal objective value. The computed bounds may still be suitable for branch-&-bound algorithms or other bounding strategies, but we do not want to evaluate the strengths and weaknesses of the discussed SDP relaxations based on rather inaccurate approximations. By the same argument, every approximate solution with an accuracy below two decimal digits was marked as 'Fail' in the corresponding tables.

Finally, we want to demonstrate the advantage of the improved cutting strategies and the introduced new SDP relaxations over competing bounding techniques which are already known in literature. For this purpose, Table 7.6 contains new best known lower bounds for selected instances from the QAP library [18]. Since the comparability to other SDP relaxations is no issue, we added some more constraints for the computation

of the bounds given in Table 7.6. These are the 2-norm constraints

$$\mathcal{L}_2(Y_\diamond) \leqslant X\,\mathcal{L}_2(B_\diamond) \qquad \text{for} \quad \diamond \in \Theta,$$

where Θ is one of the subindex sets $\{1,2\}$, $\{+,-\}$, and $\{\triangle,\triangledown\}$, respectively. For improved computing times, we furthermore removed the eigenvalue related inequalities as well as the 1-norm constraints from the MS_2 relaxations, since these had no effect on the lower bounds for the *Taixxb* instances. For the same reason, we did not apply the *EVB* based inequalities in MLX_2^* or the 1-norm cuts in $IIMS_2^*$.

Table 7.6.: New best known bounds for selected instances of the QAP library

Problem	Old bound	New bound	$R_{\text{gap}}(\%)$	CPU	Relaxation
Sko81	86 072	86 084	5.40	2655	$IIMS_2^*$
Sko90	109 030	109 529	5.20	4467	$IIMS_2^*$
Sko100a	143 846	144 443	4.97	7676	$IIMS_2^*$
Sko100b	145 522	146 128	5.04	7781	$IIMS_2^*$
Sko100c	139 881	140 619	4.90	8076	$IIMS_2^*$
Sko100d	141 289	141 861	5.16	8391	$IIMS_2^*$
Sko100e	140 893	141 673	5.01	7886	$IIMS_2^*$
Sko100f	140 691	141 295	5.19	7954	$IIMS_2^*$
Tai40b	564 428 353	574 178 590	9.90	201	MS_2^*
Tai50b	395 543 467	401 350 382	12.53	104	MS_2^*
Tai60a	5 578 356	5 596 911	22.33	1167	MLX_2^*
Tai60b	542 376 603	551 707 169	9.29	236	MS_2^*
Tai80b	717 907 288	727 622 997	11.09	946	MS_2^*
Tai100a	15 844 731	15 881 008	24.56	20 378	MLX_2^*
Tai100b	1 058 131 796	1 083 089 734	8.68	3311	MS_2^*
Tai150b	441 786 736	449 903 397	9.81	28 259	MS_2^*
Wil100	264 442	265 044	2.93	10 969	$IIMS_2$

For 17 out of 32 instances from the QAP library [19] that have not been solved to optimality yet, Table 7.6 presents new best known bounds. Each of these bounds were computed with a low-dimensional SDP relaxation. Moreover, every discussed relaxation approach for the design of a low-dimensional SDP framework is involved in the computations.

Chapter 8.

Conclusion

This thesis contributes to the topic of semidefinite programming relaxations for the quadratic assignment problem by presenting new relaxation concepts and discussing various ways to improve the respective bounding programs. These improvements primarily comprise the application of the reformulation techniques presented in Chapter 4 as well as the integration of different cutting strategies discussed in Chapter 6. In Section 7.4, we have shown that the newly introduced low-dimensional relaxation concepts are competitive to the much more expensive eigenspace splitting approach. Moreover, we have seen that the incorporation of the respective low-dimensional cutting strategies into the high-dimensional vector lifting based relaxation leads to a framework that gives good quality bounds for the corresponding problem instances.

A strong focus was also on the questions of implementation. We have seen that the attempt to model tight, low-dimensional relaxations for the computation of good quality bounds often results in ill-conditioned semidefinite programming relaxations. Even with all our attention focused on this circumstance - including the absolution of ill-posedness, the rescaling of the affected optimization variables, and the different strategies for obtaining beneficial sparsity structures in the respective programming model -, there are still some quadratic assignment problem instances for which the final relaxation frameworks are not numerically stable. However, it is important to notice that the situation would be a great deal worse without the various considerations in Chapter 7. Moreover, as pointed out in Section 7.4, these numerical problems can be resolved by applying moderate approximations. The reason why we did not use the described approximation procedure was given in the same section.

Quadratic assignment problems were, are, and most likely will remain to be optimization problems which are extremely difficult to solve, at least for the next few years. This

thesis does not chance this. However, our results improve the situation, and in some aspects they do so quite substantially. The modeled relaxation frameworks are sufficiently stable for the application in branch-&-bound procedures. Moreover, the computed lower bounds provide new bounding records for some difficult problem instances from the quadratic assignment problem library [18].

8.1. Future prospects

The search for improvements of solving strategies is something that never ends. Not surprisingly, there are several areas which seem promising for future researches and further improvements of the presented results.

We opened Section 6.2 with the claim that the corresponding frameworks would be *finalized* by the implementation of their level-2 versions. In a strict sense, this claim has already been falsified by the application of further reformulation techniques and the minimal face representation discussed in Subsection 7.2.3 and 7.2.4. But even in a wider sense, the use of the verb *finalize* is limited to the examinations in this thesis. Neither did we discuss the incorporation of the so-called *triangle inequalities*, nor did we consider the application of the *reformulation-linearization technique* [98]. The author believes that the mentioned techniques are too expensive for practical usage. Nevertheless, the investigation of these techniques may be interesting from a theoretical point of view.

Another possible future direction is the investigation of problem specific reformulations and constraints. Good examples for the exploitation of problem specific properties are Karish and Rendl's triangle decomposition approach for metric QAPs [56], Mittelmann and Pengs's SDP relaxation for quadratic assignment problems associated with a Hamming or Manhatten distance matrix [73], as well as Klerk and Sotirov's exploitation of group symmetries [24]. In our tests, we also played with different property and constraint based matrix splittings, respectively. For certain combinations of splitting approaches and specific QAP instances, we observed promising bounding improvements. It is planned to continue the research in this area. We hope to come up with a new strong low-dimensional relaxation framework.

If the applied solver implements an interior-point algorithm, solving VL_2 instances for dimensions $n \gtrsim 30$ becomes very expensive. It seems to be an interesting idea to attack this relaxation with Burer and Monteiro's low-rank factorization approach

[13, 14]. Their algorithm is designed for solving low-rank semidefinite programming problems. Unfortunately, their implementation does not handle a large number of linear programming inequalities so well. The relaxation VL_2, however, demonstrates a sensible incorporation of a relatively small number of beneficial inequality constraints. We believe that minor modifications in the relaxation framework as well as Burer and Monteiro's algorithm can lead to a bounding method for the QAP which enables us to obtain new strong lower bounds for QAPs of size $n \lesssim 80$.

Besides the respective relaxation techniques, also the practical applicability of these and the specific implementations deserve additional attention in future projects. In order to evaluate the former, we implemented a simple branch-&-bound procedure. However, the results were somehow discouraging for QAPs of sizes $n \gtrsim 40$. The growing behavior for the lower bounds seemed really bad. For the tested instances, the incorporation of additional constraints with problem specific modifications worked significantly better. Further investigations are needed to evaluate different branching strategies as well as the requirements for applicability in branch-&-bound procedures.

In Subsection 7.2.5, we demonstrated the applicability of verifications methods for the computation of rigorous bounds. Indeed, the presented numerical results illustrate the possibility of computing tight verified bounds for the respective optimal objective value with a very modest increase of the computational costs. We are currently working on extending the corresponding verification code for all other presented relaxation frameworks. Moreover, we believe that similar verification methods can be realized completely via reformulation and inclusion automatisms. A possible future prospect would be the implementation or extension of a mathematical programming language which makes use of similar verification approaches. This language should be able to handle different types of uncertainties in the input data and allow the computation of verified inclusions of the corresponding solutions.

Appendix A.

sdprQAP Quick Reference

The software package sdprQAP is written in MATLAB version 7.13. It should be compatible to all later versions of MATLAB and was also tested on Octave version 4.0. In order to use sdprQAP, it is required to install the modeling toolbox YALMIP as well as a suitable and supported solver. Compatible versions of YALMIP are Release 20150626 and Release 2015018. For the installation of sdprQAP, it is sufficient to add the package directory and its subdirectories to the MATLAB search path.

Apart from the implementations `vqapreform` and `vsdprqap` for verified computations - which are still supporting only a small range of the functionality of their counterparts `qapreform` and `sdprqap` -, Table A.1 lists all functions and scripts of sdprQAP which are relevant for the user.

The essential function for lower bound computations is `sdprqap`. It implements the functionality to set up any of the presented SDP relaxations for the QAP. The respective options are passed via a settings structure that can be generated with the function `sdprqapsettings`. If, for example, one would like to test the framework *MLX* with an approximation tolerance of 10% and without *GLB* based cuts, it is possible to use these functions as in the following code example:

```
% QAP instance ( A, B, C ) is in workspace
opts = sdprqapsettings( ...
    'framework', 'MLX', ...
    'threshold', .1, ...
    'glb', false, ...
  );
lbound = sdprqap( A, B, C, opts );
```

Table A.1.: User functions and scripts in sdprQAP

Function	Description
`BBtree`	Object class with different functions for the implementation of a branch-&-bound algorithm.
`bnbscr`	Script for a simple branch-&-bound implementation.
`evclusterapprox`	Implements the eigenvalue clustering algorithm that is used for the framework *ESC*.
`lsmatbnd`	Computes a lower sum-matrix bound for a masked, square input matrix.
`qaplibbench`	Simple benchmark script for instances from the QAPLIB.
`qapreduce`	Reduces a given QAP with additional constraints $X_{ij} = 1$ and $X_{kl} = 0$ to an equivalent QAP of smaller size.
`qapreform`	Reformulates a QAP depending on the settings.
`readqap`	Import function for problem instances given in the format used by the QAPLIB authors.
`readqapsln`	Similar to `readqap`, but for the import of a best-known assignment from the corresponding '.sln' files.
`sdprqap`	Contains implementations for all presented relaxation frameworks, including the respective cutting strategies.
`sdprqapsettings`	Generates a settings object for the use of `sdprqap`. All configuration of the applied relaxation framework is done via the returned settings structure.
`spaprojmat`	Creates sparse projection matrix to a given constant nullspace.

Table A.2.: Selected fields of sdprQAP settings structure

Field	Range	Description
`evb`	[0, 2]	Mode for *EVB* related cuts.
`framework`	string	Label for selected relaxation framework, 'ES', 'ESC', 'IIMS', 'ML', 'MLX', 'MS', 'VL', ...
`glb`	boolean	Enable/disable *GLB* based cuts.
`lpb`	[0, 2]	Mode for sum-matrix bound inequalities.
`normb`	[0, 4]	Mode for p-norm cuts.
`qapreform`	[−3, 3]	Setup of QAP reformulation.
`sdpmain`	struct	YALMIP settings structure to set options for the applied SDP solver.
`sdpsolver`	string	If not set via `sdpmain`, this option can be used to select the applied SDP solver.
`threshold`	[0.0, 1.0]	Threshold for approximation of SDP constraints.
`tolerance`	[0.0, 0.1]	General error tolerance for approximate computations in preparation code.
`weightboost`	[1.0, 10.0]	Exponent for computation of weighting coefficients used for QAP reformulation.

The variable `lbound` then contains the objective value computed by applying the respective instance of *MLX*. If requested, `sdprqap` also returns the computed approximation for the variable X as well as the diagnostics from YALMIP. A list of the most relevant options in sdprQAP is displayed in Table A.2.

A more detailed description of possible configurations and the usage of each function is given in the corresponding function documentation. For a complete list of all options in `sdprqapsettings`, type

```
help sdprqapsettings
```

into the MATLAB command window. The same procedure applies to the other functions listed in Table A.1. Alternatively, the author recommends to look into the benchmark script `qaplibbench` to get a rough feeling for the general use of the respective functions.

Constants & Sets

e	vector with all components one
E	matrix with all components one
e_i	i-th standard basis vector
I	identity matrix
0	vector/matrix with all components zero
$\mathcal{D}$	set of doubly stochastic matrices
$\mathcal{E}$	set of matrices which column and row sums are equal to one
$\mathcal{F}$	feasible set to some or all variables of a given programming problem
$\mathcal{M}$	space of real matrices
$\mathcal{N}$	cone of matrices with nonnegative components only
$\mathcal{Q}$	set of matrices with orthogonal columns
Π	set of permutation matrices
$\mathbb{R}_+$	set of nonnegative real numbers
$\mathcal{S}$	space of symmetric matrices
$\mathcal{S}_+$	cone of symmetric positive semidefinite matrices
$\mathcal{S}_{++}$	cone of symmetric positive definite matrices

Notation

$v \prec w$	v is majorized by w
$A \geqslant B$	element-wise inequality: $A - B \in \mathcal{N}$
$A \succeq B$	Löwner's partial ordering: $A - B \in \mathcal{S}_+$
$\operatorname{diag}(A)$	vector consisting of diagonal entries of A
$\operatorname{diag}^*(v)$	diagonal matrix to the components of v
$\dim(\mathcal{V})$	dimension of vector space $\mathcal{V}$
$\lambda(A)$	vector of eigenvalues of A
$\operatorname{mat}(v)$	$n \times n$ matrix to vector $v \in \mathbb{R}^{n^2}$ (column-wise indexing)
$\|\cdot\|$	Euclidean norm
$\|\|\|\cdot\|\|\|_{\mathrm{F}}$	Frobenius norm
$\|\|\|\cdot\|\|\|_2$	spectral norm
$\|\|\|\cdot\|\|\|_{\mathrm{T}}$	trace norm
$\mathcal{L}_p(A)$	vector consisting of the p-norms to the rows of A
$\operatorname{null}(\mathcal{A})$	constant nullspace to all matrices in the set $\mathcal{A}$
$\operatorname{off}(A)$	vector containing all off-diagonal components of A
$A \circ B$	Hadamard Product: $A \circ B = (a_{ij} b_{ij})$
$A \otimes B$	Kronecker Product: $A \otimes B = (a_{ij} B)$
$\langle A, B \rangle$	trace inner product: $\langle A, B \rangle = \operatorname{tr}(A^T B)$

$A^\dagger$	Moore-Penrose pseudoinverse of A
$\mathcal{R}(A)$	range, column space of A
$\text{rank}(A)$	rank of A
$\sigma(A)$	vector of singular values of A
$\text{tr}(A)$	trace of A
$\text{tri}(A)$	vector containing strict lower triangular elements of A (column-wise)
$\text{vec}(A)$	vector obtained via column-wise vectorization of A

Bibliography

[1] Albert, A. Conditions for positive and nonnegative definiteness in terms of pseudoinverses. *SIAM Journal of Applied Mathematics (SIAP) 17*, 2 (1969), 434–440.

[2] Alizadeh, F., Haeberly, J.-P. A., and Overton, M. L. Complementarity and nondegeneracy in semidefinite programming. *Mathematical Programming 77*, 1 (1997), 111–128.

[3] Assad, A. A., and Xu, W. On lower bounds for a class of quadratic 0, 1 programs. *Operations Research Letters (ORL) 4*, 4 (1985), 175–180.

[4] Berkelaar, M., Eikland, K., and Notebaert, P. lp_solve, version 5.5.2.0, 2013. Open source (Mixed-Integer) Linear Programming system, `http://lpsolve.sourceforge.net/5.5/`.

[5] Bhatia, R. *Matrix Analysis*, vol. 169 of *Graduate Texts in Mathematics*. Springer New York, 1997.

[6] Bierwirth, C., Mattfeld, D. C., and Kopfer, H. On permutation representations for scheduling problems. *Lecture Notes in Computer Science (LNCS)* (1996), 310–318.

[7] Birkhoff, G. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucumán Revista, Serie A 5* (1946), 147–151.

[8] Bolte, J., Nguyen, T. P., Peypouquet, J., and Suter, B. From error bounds to the complexity of first-order descent methods for convex functions. *ArXiv e-prints* (2015).

[9] Borchers, B. CSDP, a C library for semidefinite programming. *Optimization Methods and Software (OMS) 11*, 1–4 (1999), 613–623.

[10] Borwein, J. M., and Wolkowicz, H. Facial reduction for a cone-convex

programming problem. *Journal of the Australian Mathematical Society (ANZIAM), Series A 30*, 3 (1981), 369–380.

[11] Borwein, J. M., and Wolkowicz, H. Regularizing the abstract convex program. *Journal of Mathematical Analysis and Applications (JMAA) 83*, 2 (1981), 495–530.

[12] Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishnan, V. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[13] Burer, S., and Monteiro, R. D. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming 95*, 2 (2003), 329–357.

[14] Burer, S., and Monteiro, R. D. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming 103*, 3 (2004), 427–444.

[15] Burer, S., and Vandenbussche, D. Solving lift-and-project relaxations of binary integer programs. *SIAM Journal on Optimization (SIOPT) 16*, 3 (2006), 726–750.

[16] Burkard, R. E., Çela, E., and Klinz, B. On the biquadratic assignment problem. In *Quadratic Assignment and Related Problems* (1994), vol. 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, AMS, pp. 117–146.

[17] Burkard, R. E., Dell'Amico, M., and Martello, S. *Assignment Problems*. Society for Industrial and Applied Mathematics Philadelphia, 2012.

[18] Burkard, R. E., Karisch, S. E., and Rendl, F. QAPLIB - a quadratic assignment problem library. *Journal of Global Optimization (JOGO) 10*, 4 (1997), 391–403.

[19] Burkard, R. E., and Stratmann, K.-H. Numerical investigations on quadratic assignment problems. *Naval Research Logistics Quarterly (NRLQ) 25*, 1 (1978), 129–148.

[20] Carraresi, P., and Malucelli, F. A new lower bound for the quadratic assignment problem. *Operations Research 40* (1992), 22–27.

[21] Çela, E. *The Quadratic Assignment Problem: Theory and Algorithms*, vol. 1 of *Combinatorial Optimization*. Springer US, 1998.

[22] Cheung, Y.-L. *Preprocessing and Reduction for Semidefinite Programming via Facial Reduction: Theory and Practice*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2013.

[23] Cheung, Y.-L., Schurr, S., and Wolkowicz, H. Preprocessing and regularization for degenerate semidefinite programs. In *Computational and Analytical Mathematics: In Honor of Jonathan Borwein's 60th Birthday*, vol. 50 of *Springer Proceedings in Mathematics & Statistics*. Springer US, 2013, pp. 251–303.

[24] de Klerk, E., and Sotirov, R. Exploiting group symmetry in semidefinite programming relaxations of the quadratic assignment problem. *Mathematical Programming 122*, 2 (2010), 225–246.

[25] de Klerk, E., Sotirov, R., and Truetsch, U. A new semidefinite programming relaxation for the quadratic assignment problem and its computational perspectives. *INFORMS Journal on Computing (IJOC) 27*, 2 (2015), 378–391.

[26] Diamond, S., and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research (JMLR)* (2016). To appear.

[27] Ding, Y., and Wolkowicz, H. A low-dimensional semidefinite relaxation for the quadratic assignment problem. *Mathematics of Operations Research (MOR) 34*, 4 (2009), 1008–1022.

[28] Dür, M., Jargalsaikhan, B., and Still, G. Genericity results in linear conic programming – a tour d'horizon. *Mathematics of Operations Research (MOR)* (2016). To appear.

[29] Eaton, J. W., Bateman, D., and Hauberg, S. *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2009.

[30] Eckart, C., and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika 1*, 3 (1936), 211–18.

[31] Edwards, C. S. A branch and bound algorithm for the Koopmans-Beckmann quadratic assignment problem. In *Combinatorial Optimization II*, vol. 13 of *Mathematical Programming*. Springer Berlin Heidelberg, 1980, pp. 35–52.

[32] Elshafei, A. N. Hospital layout as a quadratic assignment problem. *Operational*

Research Quarterly (ORQ) 28, 1 (1977), 167–179.

[33] Eschermann, B., and Wunderlich, H.-J. Optimized synthesis of self-testable finite state machines. In *Fault-Tolerant Computing: 20th International Symposium (FFTCS 20)* (1990), Institute of Electrical & Electronics Engineers (IEEE), pp. 390–397.

[34] Finke, G., Burkard, R. E., and Rendl, F. Quadratic assignment problems. *Annals of Discrete Mathematics 31* (1987), 61–82.

[35] Frieze, A. M., and Yadegar, J. On the quadratic assignment problem. *Discrete Applied Mathematics (DAM) 5*, 1 (1983), 89–98.

[36] Gahinet, P., and Nemirovski, A. The projective method for solving linear matrix inequalities. *Mathematical Programming 77*, 1 (1997), 163–190.

[37] Gilmore, P. C. Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal of Applied Mathematics (SIAP) 10*, 2 (1962), 305–313.

[38] GNU Linear Programming Kit, version 4.57. `http://www.gnu.org/software/glpk/glpk.html`, 2015.

[39] Goh, J., and Sim, M. Robust optimization made easy with ROME. *Operations Research 59*, 4 (2011), 973–985.

[40] Grant, M., and Boyd, S. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, V. D. Blondel, S. P. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences. Springer London, 2008, pp. 95–110.

[41] Grant, M., and Boyd, S. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, 2015.

[42] Greenberg, H. A quadratic assignment problem without column constraints. *Naval Research Logistics Quarterly (NRLQ) 16*, 3 (1969), 417–421.

[43] Gurobi Optimization, Inc. Gurobi Optimizer v6.5. `http://www.gurobi.com`, 2015.

[44] Hadley, S. W., Rendl, F., and Wolkowicz, H. A new lower bound via projection for the quadratic assignment problem. *Mathematics of Operations Research (MOR) 17*, 3 (1992), 727–739.

[45] Hardy, G. H., Littlewood, J. E., and Pólya, G. *Inequalities*. Cambridge University Press, New York, NY, USA, 1934.

[46] Härter, V., Jansson, C., and Lange, M. VSDP: A Matlab toolbox for verified semidefinite-quadratic-linear programming. `http://www.ti3.tuhh.de/jansson/vsdp/`, 2012.

[47] Hoffman, A. J. On approximate solutions of systems of linear inequalities. *Journal of Research of the National Bureau of Standard (NIST) 49*, 4 (1952), 263–265.

[48] Horn, R. A., and Johnson, C. R. *Matrix Analysis*, 2nd ed. Cambridge University Press, New York, NY, USA, 2012.

[49] IBM ILOG CPLEX Optimization Studio, version 12.6.2. `http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud`, 2015.

[50] Ivanov, I. D., and de Klerk, E. Parallel implementation of a semidefinite programming solver based on CSDP on a distributed memory cluster. *Optimization Methods and Software (OMS) 25*, 3 (2010), 405–420.

[51] Jain, A. K., and Dubes, R. C. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[52] Jansson, C. Rigorous lower and upper bounds in linear programming. *SIAM Journal on Optimization (SIOPT) 14*, 3 (2004), 914–935.

[53] Jansson, C. On verified numerical computations in convex programming. *Japan Journal of Industrial and Applied Mathematics (JJIAM) 26*, 2–3 (2009), 337–363.

[54] Jansson, C., Chaykin, D., and Keil, C. Rigorous error bounds for the optimal value in semidefinite programming. *SIAM Journal on Numerical Analysis (SINUM) 46*, 1 (2008), 180–200.

[55] Karisch, S. E., Çela, E., Clausen, J., and Espersen, T. A dual framework for lower bounds of the quadratic assignment problem based on linearization. *Computing 63*, 4 (1999), 351–403.

[56] Karisch, S. E., and Rendl, F. Lower bounds for the quadratic assignment problem via triangle decompositions. *Mathematical Programming 71*, 2 (1995), 137–151.

[57] Kaufman, L., and Broeckx, F. An algorithm for the quadratic assignment problem using Bender's decomposition. *European Journal of Operational Research (EJOR) 2*, 3 (1978), 207–211.

[58] Keil, C., and Jansson, C. Computational experience with rigorous error bounds for the NETLIB linear programming library. *Reliable Computing 12*, 4 (2006), 303–321.

[59] Koopmans, T. C., and Beckmann, M. Assignment problems and the location of economic activities. *Econometrica 25*, 1 (1957), 53–76.

[60] Lawler, E. L. The quadratic assignment problem. *Management Science 9*, 4 (1963), 586–599.

[61] Li, X. S., Martin, M. C., Thompson, B. J., Tung, T., Yoo, D. J., Demmel, J. W., Bailey, D. H., Henry, G., Hida, Y., Iskandar, J., Kahan, W., Kang, S. Y., and Kapur, A. Design, implementation and testing of extended and mixed precision BLAS. *ACM Transactions on Mathematical Software (TOMS) 28*, 2 (2002), 152–205.

[62] Li, Y., Pardalos, P. M., Ramakrishnan, K. G., and Resende, M. G. C. Lower bounds for the quadratic assignment problem. *Annals of Operations Research (AOR) 50*, 1 (1994), 387–10.

[63] Lidskiĭ, V. B. The proper values of the sum and product of symmetric matrices. *Doklady Akademii nauk SSSR 75* (1950), 769–772.

[64] Löfberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference* (2004).

[65] Löfberg, J. Dualize it: software for automatic primal and dual conversions of conic programs. *Optimization Methods and Software (OMS) 24*, 3 (2009), 313–325.

[66] Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T. A survey for the quadratic assignment problem. *European Journal of Operational Research (EJOR) 176*, 2 (2007), 657–690.

[67] Lourenço, B. F., Muramatsu, M., and Tsuchiya, T. Solving SDP completely with an interior point oracle. *ArXiv e-prints* (2015).

[68] Lourenço, B. F., Muramatsu, M., and Tsuchiya, T. A structural geometrical analysis of weakly infeasible SDPs. *ArXiv e-prints* (2015).

[69] Löwner, K. Über monotone Matrixfunktionen. *Mathematische Zeitschrift 38*, 1 (1934), 177–216.

[70] Macdonald, I. G. *Symmetric Functions and Hall Polynomials*, 2 ed. The Clarendon Press, Oxford University Press, New York, NY, USA, 1995.

[71] Mangasarian, O. L. A condition number for linear inequalities and equalities. In *Methods of Operations Research: Proceedings of the 6th Symposium on Operations Research* (1981), G. Bamberg and O. Opitz, Eds., vol. 43, Athenäum-Verlag, pp. 3–15.

[72] Mirsky, L. Symmetric gauge functions and unitarily invariant norms. *Quarterly Journal of Mathematics 11*, 1 (1960), 50–59.

[73] Mittelmann, H., and Peng, J. Estimating bounds for quadratic assignment problems associated with Hamming and Manhattan distance matrices based on semidefinite programming. *SIAM Journal on Optimization (SIOPT) 20*, 6 (2010), 3408–3426.

[74] Moore, E. H. On the reciprocal of the general matrix. *Bulletin of the American Mathematical Society (AMS) 26* (1920), 394–395.

[75] MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual, version 7.1 (revision 28). `http://docs.mosek.com/7.1/toolbox/index.html`, 2015.

[76] NETLIB linear programming library. A collection of linear programming problems, `http://www.netlib.org/lp/`.

[77] O'Donoghue, B., Chu, E., Parikh, N., and Boyd, S. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications (JOTA)* (2016). To appear.

[78] Ogita, T., Rump, S. M., and Oishi, S. Accurate sum and dot product. *SIAM Journal on Scientific Computing (SISC) 26*, 6 (2005), 1955–1988.

[79] Ozaki, K., Ogita, T., Oishi, S., and Rump, S. M. Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications. *Numerical Algorithms 59*, 1 (2011), 95–118.

[80] Pardalos, P. M., Rendl, F., and Wolkowicz, H. The quadratic assignment problem: A survey and recent developments. In *Quadratic Assignment and Related Problems* (1994), vol. 16 of *DIMACS Series in Discrete Mathematics and Theoretical*

Computer Science, American Mathematical Society (AMS), pp. 1–42.

[81] Pataki, G. Strong duality in conic linear programming: facial reduction and extended duals. In *Computational and Analytical Mathematics: In Honor of Jonathan Borwein's 60th Birthday*, vol. 50 of *Springer Proceedings in Mathematics & Statistics*. Springer New York, 1980, pp. 613–634.

[82] Pataki, G., and Tunçel, L. On the generic properties of convex optimization problems in conic form. *Mathematical Programming 89*, 3 (2001), 449–457.

[83] Peng, J., Mittelmann, H., and Li, X. A new relaxation framework for quadratic assignment problems based on matrix splitting. *Mathematical Programming Computation (MPC) 2*, 1 (2010), 59–77.

[84] Peng, J., Zhu, T., Luo, H., and Toh, K.-C. Semi-definite programming relaxation of quadratic assignment problems based on nonredundant matrix splitting. *Computational Optimization and Applications (COAP) 60*, 1 (2014), 171–198.

[85] Penrose, R., and Todd, J. A. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society 51*, 3 (1955), 406–413.

[86] Povh, J., and Rendl, F. Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization 6*, 3 (2009), 231–241.

[87] Rendl, F., and Sotirov, R. Bounds for the quadratic assignment problem using the bundle method. *Mathematical Programming 109*, 2–3 (2007), 505–524.

[88] Rendl, F., and Wolkowicz, H. Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem. *Mathematical Programming 53*, 1–3 (1992), 63–78.

[89] Renegar, J. Incorporating condition measures into the complexity theory of linear programming. *SIAM Journal on Optimization (SIOPT) 5*, 3 (1995), 506–524.

[90] Robinson, S. M. Bounds for error in the solution set of a perturbed linear program. *Linear Algebra and its Applications (LAA) 6* (1973), 69–81.

[91] Roucairol, C. Un nouvel algorithme pour le problème d'affectation quadratique. *RAIRO-Recherche Opérationnelle - Operations Research 13*, 3 (1979), 275–301.

[92] Rump, S. M. On the solution of interval linear systems. *Computing 47*, 3–4 (1992), 337–353.

[93] Rump, S. M. INTLAB - INTerval LABoratory. In *Developments in Reliable Computing*, T. Csendes, Ed. Kluwer Academic Publishers, 1999, pp. 77–104.

[94] Rump, S. M. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica 19* (2010), 287–449.

[95] Rump, S. M., and Kaucher, E. Small bounds for the solution of systems of linear equations. In *Computing Supplementum*. Springer Vienna, 1980, pp. 157–164.

[96] Sahni, S., and Gonzalez, T. P-complete approximation problems. *Journal of the Association for Computing Machinery (JACM) 23*, 3 (1976), 555–565.

[97] Schur, I. Über eine Klasse von Mittelbildungen mit Anwendungen auf die Determinanten-Theorie. *Sitzungsberichte der Berliner Mathematischen Gesellschaft (BMG) 22* (1923), 9–20.

[98] Sherali, H. D., and Adams, W. P. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, vol. 31 of *Nonconvex Optimization and Its Applications*. Springer US, 1999.

[99] Slater, M. Lagrange multipliers revisited. In *Traces and Emergence of Nonlinear Programming*. Springer Basel, 2013, pp. 293–306.

[100] Steinberg, L. The backboard wiring problem: A placement algorithm. *SIAM Review (SIREV) 3*, 1 (1961), 37–50.

[101] Sturm, J. F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software (OMS) 11–12* (1999), 625–653.

[102] Toh, K.-C., Todd, M. J., and Tütüncü, R. H. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, M. F. Anjos and J. B. Lasserre, Eds., vol. 166 of *International Series in Operations Research and Management Science*. Springer US, 2011, pp. 715–754.

[103] Tütüncü, R. H., Toh, K.-C., and Todd, M. J. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming 95*, 2 (2003), 189–217.

[104] Udell, M., Mohan, K., Zeng, D., Hong, J., Diamond, S., and Boyd, S. Convex optimization in Julia. *SC14 Workshop on High Performance Technical Computing in Dynamic Languages* (2014).

[105] von Neumann, J. Some matrix-inequalities and metrization of matrix-space. *Tomsk University Review 1* (1937), 286–300.

[106] Waki, H., and Muramatsu, M. Facial reduction algorithms for conic optimization problems. *Journal of Optimization Theory and Applications (JOTA) 158*, 1 (2012), 188–215.

[107] Weyl, H. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematischen Annalen (MA) 71*, 4 (1912), 441–479.

[108] Wolkowicz, H. Semidefinite programming approaches to the quadratic assignment problem. In *Nonlinear Assignment Problems: Algorithms and Applications*, P. M. Pardalos and L. S. Pitsoulis, Eds., vol. 7 of *Combinatorial Optimization*. Springer US, 2000, pp. 143–174.

[109] Xia, Y. Gilmore-lawler bound of quadratic assignment problem. *Frontiers of Mathematics in China 3*, 1 (2008), 109–118.

[110] Xia, Y. Second order cone programming relaxation for quadratic assignment problems. *Optimization Methods and Software (OMS) 23*, 3 (2008), 441–49.

[111] Xia, Y., and Yaun, Y.-X. A new linearization method for quadratic assignment problems. *Optimization Methods and Software (OMS) 21*, 5 (2006), 805–818.

[112] Yamashita, M., Fujisawa, K., Fukuda, M., Kobayashi, K., Nakata, K., and Nakata, M. Latest developments in the SDPA family for solving large-scale SDPs. In *Handbook on Semidefinite, Conic and Polynomial Optimization*. Springer US, 2011, pp. 687–713.

[113] Yang, L., Sun, D., and Toh, K.-C. SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Mathematical Programming Computation (MPC) 7*, 3 (2015), 331–366.

[114] Zhang, F., Ed. *The Schur Complement and Its Applications*, vol. 4 of *Numerical Methods and Algorithms*. Springer US, 2005.

[115] Zhang, S. Global error bounds for convex conic problems. *SIAM Journal on Optimization (SIOPT) 10*, 3 (2000), 836–851.

[116] Zhao, Q., Karisch, S. E., Rendl, F., and Wolkowicz, H. Semidefi-

nite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization (JOCO) 2*, 1 (1998), 71–109.

[117] Zhao, X.-Y., Sun, D., and Toh, K.-C. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization (SIOPT) 20*, 4 (2010), 1737–1765.

[118] Zhu, Y.-K., and Hayes, W. B. Algorithm 908: Online exact summation of floating-point streams. *ACM Transactions on Mathematical Software (TOMS) 37*, 3 (2010), 1–13.

Curriculum Vitae

Personal Information	
Name	Marko Lange
Nationality	German
Date of birth	01/04/1985
Place of birth	Rostock, Germany
Gender	Male
Primary Education	
08/1991 - 06/1995	Grundschule Alter Markt, Rostock
Secondary Education	
08/1995 - 06/2004	Gymnasium Große Stadtschule, Rostock
Civilian Service	
09/2004 - 05/2005	Kath. Marienkrankenhaus gGmbH Hamburg
Tertiary Education	
10/2005 - 01/2012	Hamburg University of Technology Course: Electrical Engineering Degree: Diplom Ingenieur
Work Experience	
01/2012 - 03/2016	Research Associate Institute for Reliable Computing Hamburg University of Technology