# Sparse Recovery Under Side Constraints Using Null Space Properties

Vom Fachbereich Mathematik
der Technischen Universität Darmstadt
zur Erlangung des Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigte

**Dissertation**

von

**Frederic Matter, M. Sc.**

aus München

| | |
|---|---|
| Referent: | Prof. Dr. Marc E. Pfetsch |
| Korreferent: | Prof. Dr. Thorsten Theobald |
| Tag der Einreichung: | 9. Februar 2022 |
| Tag der mündlichen Prüfung: | 29. April 2022 |

Darmstadt 2022

# Acknowledgments

First and foremost, I would like to thank my supervisor Marc E. Pfetsch. He gave me the great opportunity to work on this thesis within the topic on Compressed Sensing. Whenever I had questions or problems, be it technical issues with SCIP(-SDP) or of theoretical nature, he was available and spent much time discussing and helping me. I am very grateful for all the discussion and the most valuable input, from which I learned and profited a lot.

Furthermore, I would like to thank Thorsten Theobald. Having finished my Master's thesis under his great supervision, he referred me to Marc, when I came up with the idea to do a PhD. I am very happy that we stayed in contact and I especially very much enjoyed working together with him. Finally, I would like to thank him for agreeing to act as a referee for this thesis.

Moreover, I would like to thank all of my colleagues within the "EXPRESS" project for the great collaboration, especially Marius Pesavento for several fruitful discussions.

Besides, I would like to thank Frank Aurzada and Marius Pesavento for agreeing to to be part of the examination committee for my thesis.

I gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) within the project "Exploiting Structure in Compressed Sensing Using Side Constraints (EXPRESS)" in the priority programme SPP 1798. If not for the funding, this thesis would not have been possible.

In that regard, I am very grateful to Dennis Gabriel, Ilhan Gören, Maximilian Gläser, Oliver Habeck, Christine Herter, Christopher Hojny, Erik Jansen, Jasmin Pabst and Andreas Schmitt for proofreading parts of this thesis and providing valuable feedback.

Something I enjoyed a lot is the great working atmosphere within the whole Optimization group at TU Darmstadt. Thanks for all the great discussions and various other activities, especially the stays at Kleinwalsertal. Special thanks go out to Christopher Hojny for taking his time and helping me whenever I encountered problems with SCIP, which happened quite often at the beginning. In particular, many thanks go to Oliver Habeck as well as Christine Herter for being such great friends and office mates, with our shared love for “your mother”-jokes. Working without you would definitely have been not half as enjoyable. We shared countless fun coffee breaks with many cookies. Oliver revolutionized my eating habits by creating Blitz-Friday. Christine always lent me an ear and cheered my up whenever I needed it, especially in the last months. Also, thanks go to Andreas Schmitt for many office chats and great laughs.

My family was always there for supporting and encouraging me whenever I needed it most—and still is. Thank you for all the love you gave to me, and of course the cakes.

Finally, my deepest gratitude goes out to Jasmin Pabst, for her neverending support, which now started almost ten years back. Especially in the last months, I became quite unnerving and even occasionally intolerable, but she still kept me going. Without your caring support, I would most certainly not be finished by now, and most likely also starved already long ago—one cannot simply eat Blitz-Pizza everyday (unfortunately).

# Zusammenfassung

Ein Kernaspekt von Compressed Sensing ist die Rekonstruktion von Signalen mithilfe von möglichst wenigen Messungen. Dies wird unter Ausnutzung der Tatsache ermöglicht, dass sich die Signale in vielen Anwendungen durch lediglich wenige Komponenten beschreiben lassen. Solch ein Signal kann mathematisch durch einen sogenannten dünnbesetzten Vektor modelliert werden. Die Messungen des Signals lassen sich mit einer Messmatrix darstellen, wobei die Anzahl an Zeilen der Messmatrix gerade der Anzahl an Messungen entspricht. Dies führt auf ein lineares Gleichungssystem, welches in der Regel unterbestimmt ist. Durch Rekonstruktionsbedingungen an die Messmatrix, wie der sogenannten „Null Space Property" (NSP), kann genau charakterisiert werden, in welchen Fällen ein dünnbesetzter Vektor aus seinen Messungen erfolgreich rekonstruiert werden kann.

In der vorliegenden Arbeit untersuchen wir dieses mathematische Problem der Rekonstruktion unter dem Aspekt, dass zusätzliche Informationen über die Struktur des zu rekonstruierenden dünnbesetzten Vektors vorhanden ist. Solche Informationen können in Form von Nebenbedingungen beschrieben werden. Als einen zentralen Punkt weisen wir anhand von verschiedenen Beispielen empirisch nach, dass schwächere Rekonstruktionsbedingungen in der Gegenwart von Nebenbedingungen möglich sind und dass weniger Messungen zur Rekonstruktion benötigt werden.

Hierzu entwickeln wir zuerst ein allgemeines Framework, welches ein existierendes Framework aus der Literatur um die Möglichkeit erweitert, vorhandene Nebenbedingungen auszunutzen. Zur Charakterisierung der erfolgreichen Rekonstruktion in diesem Framework formulieren wir eine neue allgemeine NSP. Weiterhin betrachten wir auch den Fall, dass die Messungen durch Rauschen gestört sind oder die zu rekonstruierenden Signale sich lediglich durch dünnbesetzte Vektoren approximieren lassen. Wir zeigen, dass eine Modifikation unserer allgemeinen NSP auch diese Fälle umfasst und formulieren Schranken an den möglichen Fehler, der bei der Rekonstruktion gemacht wird.

Das vorgestellte Framework vereinigt und verallgemeinert diverse in der Literatur behandelten Spezialfälle, wie zum Beispiel die Rekonstruktion von dünnbesetzten, nichtnegativen oder ganzzahligen Vektoren, dünnbesetzten Vektoren mit Block-

Struktur oder positiv semidefiniten Matrizen mit niedrigem Rang. Wir demonstrieren, dass sich die bekannten Resultate für diese Spezialfälle aus unserem allgemeinen Framework herleiten lassen. Des Weiteren betrachten wir auch noch weitere Spezialfälle, die in der Literatur bisher nicht oder nur wenig untersucht wurden. Zuerst erweitern wir eine mögliche Block-Struktur in Vektoren auf Matrizen. Hier gehen wir auch explizit auf den Effekt von Nichtnegativität oder positiver Semidefinitheit ein und zeigen anhand von Beispielen, dass durch Ausnutzung dieser Eigenschaften eine schwächere Rekonstruktionsbedingung nötig ist. Danach untersuchen wir ganzzahlige Vektoren und schließlich behandeln wir komplexe Vektoren, bei denen jeder Eintrag einen konstanten Betrag hat. Für die letztgenannte Nebenbedingung präsentieren wir auch einen angepassten „Spatial Branch-and-Bound“ Algorithmus zur Lösung des entstehenden Rekonstruktionsproblems.

Um den Effekt von Nichtnegativität näher zu analysieren, betrachten wir die NSP für zufällige Messmatrizen. Wir leiten eine theoretische untere Schranke für die Anzahl an nötigen Messungen her und vergleichen diese empirisch und numerisch mit der bekannten Schranke im Falle von dünnbesetzten Vektoren ohne Nichtnegativität. Dies demonstriert, dass unter Ausnutzung der Nichtnegativität weniger Messungen zur Rekonstruktion nötig sind. Ebenfalls leiten wir eine solche Schranke für Matrizen mit Block-Struktur her.

Anschließend behandeln wir dünnbesetzte Vektoren mit und ohne Nichtnegativität sowie mit und ohne Block-Struktur. Für diese Fälle formulieren wir das Problem, die jeweilige NSP für eine gegebene Messmatrix zu überprüfen, als gemischt-ganzzahliges Optimierungsproblem. Der Effekt der Nichtnegativität zeigt sich hierbei darin, dass das zugehörige Optimierungsproblem zum Nachweisen der NSP schneller gelöst werden kann. Empirisch weisen wir auch nach, dass in der Gegenwart von Nichtnegativität weniger Messungen, das heißt Zeilen in einer zufälligen Messmatrix, nötig sind, um die NSP zu erfüllen.

Für die „Restricted Isometry Property“ (RIP), welche eine weitere Rekonstruktionsbedingung für dünnbesetzte Vektoren darstellt, betrachten wir die Formulierung als gemischt-ganzzahliges semidefinites Optimierungsproblem (MISDP). Dies führt uns schließlich zu allgemeinen MISDPs. Wir entwickeln neue Techniken zur Modifikation von MISDPs vor und während dem Lösen, um dadurch den Lösungsprozess zu beschleunigen. Anhand von numerischen Ergebnissen auf verschiedenen Klassen von MISDPs sehen wir, dass dadurch eine signifikante Beschleunigung möglich ist. Hierbei legen wir ein besonderes Augenmerk auf die MISDP Formulierung der RIP.

# Abstract

A key aspect of Compressed Sensing is the reconstruction of signals with as few measurements as possible. This can be achieved by exploiting that in many applications, signals can be described using only few components, which results in so-called sparse vectors. The measurements of a signal can be represented with a measurement matrix whose number of rows is exactly the number of measurements taken. This leads to a linear equation system, which typically is underdetermined. In order to characterize when a sparse vector can successfully be reconstructed from its measurements, so-called reconstruction guarantees such as the “Null Space Property” (NSP) can be employed.

This thesis examines sparse recovery in the case that additional structure is known in the sparse vector that is to be recovered. As one key point, it is empirically demonstrated that in the presence of side constraints weaker recovery guarantees are possible and that fewer measurements suffice for successful recovery. To do so, a general framework for sparse recovery is developed, which allows to incorporate additional knowledge in form of side constraints and a novel general NSP is proposed, which characterizes successful recovery in this framework. This framework subsumes many specific settings and NSPs already considered in the literature.

For the case of sparse vectors, the influence of nonnegativity is analyzed by considering whether random measurement matrices satisfy the corresponding NSPs. A lower bound for the number of measurements needed for successful recovery is derived and empirically as well as numerically compared to the known bound for sparse vectors without nonnegativity. Afterwards, the problem of testing whether a given measurement matrix satisfies an NSP is considered. For the explicit cases of sparse (nonnegative) vectors and block-sparse (nonnegative) vectors, the problem of testing the corresponding NSP is formulated as a mixed-integer problem. Empirical results demonstrate that for a random measurement matrix, fewer measurements are needed in order to satisfy the corresponding NSP in the presence of nonnegativity.

Lastly, new presolving and propagation techniques for general mixed-integer semidefinite programs (MISDPs) are developed, which allow for a significant improvement in the solution times, as a numerical evaluation on several classes of

MISDPs reveals. In this computational study, a focus lies on the MISDP formulation of the “Restricted Isometry Property” (RIP), which is another recovery guarantee for sparse vectors.

# Contents

# CHAPTER 1

# Introduction

Our modern world demands the use of digital data and information in almost every aspect of our everyday life. Transferring, measuring and reconstructing this data or information is therefore an omnipresent task. For instance, in digital communication, signals or images frequently need to be reconstructed from measured data. Measuring, or acquiring data in its simplest form amounts to a system of linear equations $Ax = b$. Here, $x \in \mathbb{R}^n$ is the original $n$-dimensional signal, $A \in \mathbb{R}^{m \times n}$ is the measurement matrix and $b \in \mathbb{R}^m$ collects the $m$ measurements taken of $x$ by $A$. Clearly, this system is underdetermined as long as $m < n$, so that if there exists any solution, it is not unique. Consequently, recovering $x$ cannot be expected. Taking $m > n$ measurements in order to hope for a unique solution is undesirable in practice, simply because the dimension $n$ of the original signal may be very large. Hence, the measurement process can become costly and time-consuming, both of which should typically be avoided. Thus, additional information on $x$ is needed in order to successfully recover $x$, that is, to have a unique solution of $Ax = b$, which is equal to $x$ even if $m < n$.

Almost two decades ago, the crucial observation was made that *sparsity* is exactly such an additional information which makes reconstruction possible. Sparsity means that the signal vector only has few nonzero components, that is, the vector $x$ modeling the signal only has few nonzero entries. We call a vector $s$-sparse, if it contains at most $s$ nonzero components. In the following, we will speak of signals and vectors interchangeably. The assumption that a signal is sparse or can be approximated by a sparse signal, holds in many real world applications, possibly after changing the representation of the signal, that is, after changing the basis. Examples include the famous JPEG, MPEG and MP3 formats for image, video and audio data, respectively. These formats compress the data by finding a sparse approximation in a suitable basis. Consequently, some information within the data is lost in the compression process. However, this loss does not significantly affect the quality of the data, but massively shrinks the necessary storage space. Utilizing such *sparse* or *compressed* representations of signals leads to the problem that first measuring a signal and then effectively throwing away most of the information when compressing the signal is clearly not necessary. Rather, measuring and compressing the signal should be done as one step, such that directly the compressed version of the signal should be acquired. This paradigm to simultaneously acquire and compress the data is now known under the name *Compressed Sensing (CS)* (or Compressive Sensing, Compressive Sampling) which started with the seminal articles from Candès et al. [39] as well as Donoho [69].

CS has numerous applications, probably the most well-known is magnetic resonance imaging (MRI) in medicine, see, e.g., Lustig et al. [166, 167]. In fact, MRI is the motivation used by Candès et al. [39], considered as the initial paper on CS. In MRI, the task is to produce high-quality images displaying the anatomy of parts of the human body. Clearly, in order to diagnose diseases such as cancer, the pictures should be as high-resolution as possible. Moreover, since the radiation used in an MRI is harmful for the human body, the exposition to it should be as short as possible. Thus, CS techniques can be used to reduce the number of measurements that need to be taken without influencing the quality of the resulting images, so that the duration of the exposition to radiation can be shortened. Apart from this medical application, CS has applications in radar frameworks, e.g., for detecting objects in the surrounding environment and measuring their distance and speed, see Herman and Strohmer [127], Potter et al. [204] and also Foucart and Rauhut [104] from which the following high-level description is borrowed. In order to detect objects, a radar pulse is sent out, which is scattered at these objects. The resulting scattered signal is then measured at a receive antenna. Taking the delay of the received signal as well as the Doppler effect into account, the distance and the speed of the objects can be computed. In a finite-dimensional model, a known channel matrix can be used to compute the received signal based on the sent signal, and a vector $x$ can be used to model the presence of objects with a certain speed and distance. This vector is typically sparse, since only few objects are present in the surrounding environment. Thus, the goal is to reconstruct a sparse vector based on linear measurements. Moreover, CS is frequently used to find *sparse approximations* of vectors in a predefined basis, also called dictionary. As outlined before, this is heavily used in compression, but also in data separation [202] and denoising of data [85], see also [35, 84] for an overview. Besides, CS can be applied in order to correct errors in the transmission of data [38].

For further information and more general introductions as well as applications of CS, we refer to the books [88, 104, 248] as well as the papers [14, 40, 46, 99]. The PhD thesis of Andreas Tillmann [235] collects several computational aspects of CS. We provide further literature reviews for the specific topics considered in Chapters 3 and 5 as well as in Sections 4.1 and 6.1 at the beginning of the respective chapters.

Deepening knowledge in the usage of CS is the primary goal of the priority programme SPP 1798 “Compressed Sensing in Information Processing (CoSIP)”, funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) from 2015 to 2021. The subproject “Exploiting Structure in Compressed Sensing Using Side Constraints (EXPRESS)” within the SPP 1798, which ran for the full period of six years, deals with the question of how additional knowledge can be exploited in the process of acquiring data. Such knowledge may be available for different aspects of the recovery process. It can originate from specific structure within the measurement matrix $A$, the original signal $x$ or the measurements $b$. For instance, it may be known that $x \geq 0$ or that $x$ has integral components. Moreover, the nonzero entries in $x$ can appear in blocks or groups, which amounts to knowledge on the sparsity structure. Depending on the measurement process, the measurements can exhibit structure as well, such as being magnitude-only, quantized, or restricted to a finite alphabet ($K$-bit measurements). Lastly, specific properties of the underlying measurement process, such as configurations of the arrays used for sensing, can yield additional structure in the measurement matrix $A$. This knowledge can be incorporated into the recovery problem in form of a side constraint. In the project “EXPRESS”, new recovery guarantees as well as efficient algorithms for recovery in the presence of side constraints are developed, with a specific focus on the application to multi-antenna systems. This thesis emerged from the research that has been done within the project “EXPRESS”, and solely deals with additional structure in $x$. An overview over various results that have been obtained in this project is given in the preprint by Ardah et al. [10].

## 1.1 Sparse Recovery Under Linear Measurements

In this thesis, we will focus on the sparse recovery problem, which consists of reconstructing an unknown sparse vector $x \in \mathbb{R}^n$ given only its measurements $b = Ax$, where $A \in \mathbb{R}^{m \times n}$ is a known measurement matrix. This problem features two important aspects: First, the choice of $A$, and second, the recovery process itself.

Concerning the choice of $A$, it is desirable to choose a measurement matrix which is suitable to reconstruct different sparse vectors, not only a fixed one. This distinction leads to the terms *individual recovery* (also called nonuniform recovery) and *uniform recovery*. Individual recovery means the successful recovery of a single fixed $s$-sparse vector $x \in \mathbb{R}^n$ from its measurements $b = Ax \in \mathbb{R}^m$, whereas for uniform recovery, all $s$-sparse vectors need to be successfully recovered from their measurements $b$ using the same measurement matrix $A$.

Most importantly, the recovery process should be efficient. An immediate idea would be to find a vector with a minimal amount of nonzero components which is compatible with the measurements. Let $\|x\|_0$ denote the number of nonzero components of the vector $x$, then this leads to the so-called $\ell_0$-minimization problem

$$\min\{\|x\|_0 \,:\, Ax = b\}, \tag{$P_0$}$$

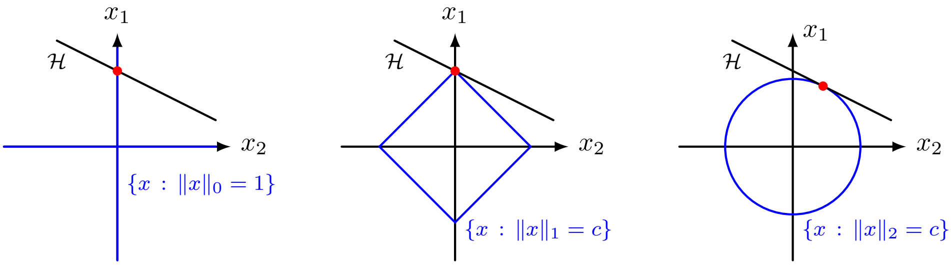
where $A \in \mathbb{R}^{m\times n}$ is a measurement matrix and $b \in \mathbb{R}^m$ are given measurements. However, Natarajan [185] shows that this problem is in general $\mathcal{NP}$-hard, and even producing approximate solutions is considered to be intractable, see Amaldi and Kann [8]. Quite surprisingly it turns out that after replacing $\|\cdot\|_0$ by the $\ell_1$-norm $\|\cdot\|_1$ it is often still possible to successfully reconstruct sparse signals from their measurements. Using $\|\cdot\|_1$ leads the $\ell_1$-minimization problem

$$\min\{\|x\|_1 \,:\, Ax = b\}, \tag{$P_1$}$$

which is also known as *basis pursuit*. It probably appeared in the work by Chen et al. [51] for the first time explicitly in sparse recovery, even if the general idea of using $\ell_1$-minimization seems to be much older, see Donoho and Logan [72] and Logan [161]. In fact, basis pursuit can be written as a linear program (LP) and thus is efficiently solvable in polynomial time. Figure 1.1 shows the intuition behind basis pursuit. For an underdetermined system of linear equations $Ax = b$, a solution with minimal $\ell_1$- or $\ell_2$-norm is given by inflating the unit ball of the respective norm until it first has contact with the affine space $\mathcal{H} = \{x \,:\, Ax = b\}$. This contact point displayed in Figures 1.1b and 1.1c, respectively, shows that the solution with minimal $\ell_1$-norm is indeed sparse, whereas the solution with minimal $\ell_2$-norm does not contain a nonzero entry. Moreover, Figure 1.1a also shows a sparse solution with minimal $\ell_0$-norm.

Besides solving basis pursuit, there are also other algorithms and recovery schemes that can be used for sparse recovery. Examples include greedy algorithms such as orthogonal matching pursuit (OMP) or compressive sampling matching pursuit (CoSaMP) [186], or thresholding algorithms such as iterative hard thresholding (IHT). Those will not be treated throughout this thesis, we will only consider basis pursuit ($P_1$). For more information on these algorithms, we refer to Foucart and Rauhut [104, Section 3] as well as Tropp [239].

The most important question regarding $\ell_1$-minimization is when it is possible to reconstruct sufficiently sparse vectors from their measurements using ($P_1$), either



**(a)** The $\ell_0$-norm. **(b)** The $\ell_1$-norm. **(c)** The $\ell_2$-norm.

**Figure 1.1.** Geometric intuition for Basis Pursuit: The intersection of an affine space $\mathcal{H} = \{x : Ax = b\}$ with the unit norm ball of the $\ell_0$-norm as well as inflated unit norm balls of the $\ell_1$- and $\ell_2$-norm for some $c > 0$.

only a fixed sparse vector (individual recover), or all sufficiently sparse vectors (uniform recovery). In order to answer this question, recovery conditions have been proposed in the literature. These impose conditions on the measurement matrix which guarantee successful recovery. The historically first condition is the restricted isometry property (RIP), introduced by Candès and Tao [38, 45]. As shown by Candès and Tao [38], the RIP is a sufficient condition for uniform recovery. Another by now well-known condition is the null space property (NSP). After appearing implicitly in works by Donoho and Elad [70], Donoho and Huo [71], Elad and Bruckstein [86], and Gribonval and Nielsen [123], the term NSP was first used by Cohen et al. [55]. In contrast to the RIP, the NSP characterizes uniform recovery, i.e., it is a necessary and sufficient condition. Further recovery conditions include conditions on the *coherence* [70, 123] or the *spark* [55, 70] of the measurement matrix. The latter leads to a condition for uniform recovery using $(P_0)$ instead of $(P_1)$. For individual recovery, there also exist specific conditions such as the exact recovery condition (ERC) of Tropp [239] or conditions based on dual certificates of $(P_1)$. Since this thesis is only concerned with the NSP and, to some extent, the RIP, we again refer to Foucart and Rauhut [104] and the references therein for more details on other recovery conditions.

Another important property is the *stability* and *robustness* of the recovery scheme. In reality, most signals are not sparse, but only close (in some distance metric) to sparse signals. Moreover, measurements are often corrupted by noise and thus inaccurate. Hence, it is important to control the reconstruction error. This is referred as stability and robustness of the recovery process. Basis pursuit can be made stable and robust if the condition $Ax = b$ is weakened to $\|Ax - b\|_2 \leq \eta$, where $\eta \geq 0$ is a known bound for the measurement noise. This leads to the following version of basis pursuit, called *basis pursuit denoising*:

$$\min\{\|x\|_1 \,:\, \|Ax-b\|_2 \le \eta\}. \tag{$P_1^\eta$}$$

In fact, the recovery conditions such as the NSP and the RIP can be adapted accordingly to also give guarantees on the recovery error of uniform stable or robust recovery. Candès et al. [39] extended the RIP to stable and robust recovery. The term *stable and robust NSP* was seemingly first used by Foucart and Rauhut [104], for conditions which appeared throughout the literature.

In the past decade, the classical CS problem of recovering sparse vectors was extended by using different types of sparsity. For instance, block-sparsity groups entries of a vector into blocks and demands that only few blocks contain nonzero entries. Moreover, the recovery problem can be generalized from vectors to matrices, which leads to low-rank matrix recovery, see, e.g., Recht et al. [210]. In [128], a block-structure on matrices was introduced which generalizes block-sparse vectors. This block-structure will be treated in more detail in Section 3.1. The NSP and other recovery conditions have been adapted to these types of sparsity in the literature as well, see Stojnic et al. [230] for block-sparse vectors, Oymak and Hassibi [192] for low-rank matrices and [128] for block-sparse matrices.

These recovery conditions have a very similar structure, and their proofs use comparable arguments. This led to different frameworks, which have been proposed in the literature to subsume the existing theory. Examples include the concept of decomposable norms used by Negahban et al. [188] and Candès and Recht [44] as well as atomic norms in Chandrasekaran et al. [49]. A very general setting that subsumes most of the existing NSPs has been introduced by Juditsky et al. [137]. These frameworks provide a convenient generalization of the explicit recovery conditions which have been derived in the literature for different settings. However, it turns out that they do not allow exploiting additional types of structure. For example, depending on the applications, it may be known in advance that a signal is represented with only a few nonnegative entries, or a positive semidefinite low-rank matrix. Such side constraints can easily be added to $\ell_1$-minimization and its adaptions. If the side constraint is convex, then the recovery problem stays tractable, at least in theory. Clearly, not exploiting side constraints is always feasible, since, e.g., uniform recovery of all sufficiently sparse vectors implies uniform recovery of all sufficiently sparse nonnegative vectors. However, it is natural to believe that explicitly imposing nonnegativity yields weaker recovery conditions and that fewer measurements are needed for uniform or individual recovery as a result.

This thesis presents results on this topic by providing a general framework which builds upon the framework by Juditsky et al. [137] and extends it to also incorporate additional knowledge in the form of side constraints. As one main result, we present general null space properties for exact, stable and robust recovery. It will turn out that the known recovery conditions in the presence of side constraints emerge as special cases from the proposed generalization.

Apart from the clear focus on general null space properties, this thesis also shortly deals with the RIP. It is known that the checking whether a given measurement matrix satisfies the RIP can be formulated as a mixed-integer semidefinite program (MISDP), see Gally and Pfetsch [111]. For solving the resulting MISDP, the software SCIP-SDP [220] can be used. In the process of trying to find measurement matrices satisfying the RIP, noticeable effort has been put into improving the performance of SCIP-SDP by Marc E. Pfetsch and the author of this thesis. In particular, several new presolving and propagation techniques have been implemented, and most of the code has been touched and revised. This has led to version 4.0 of SCIP-SDP, see also [25]. We refer to Section 6.1 for more information and an introduction to SCIP-SDP.

## 1.2 Outline and Contribution

The main goal of this thesis is to analyze sparse recovery under additional side constraints, as one of the main research topics in the project “EXPRESS” within the SPP 1798. As described in the previous section, we solely consider $\ell_1$-minimization and its variants in different settings, and are interested in deriving adjusted recovery guarantees for the resulting recovery problem in these settings. Our primary focus is on null space properties.

In Chapter 2, we extend the general framework for sparse recovery using a generalized version of $\ell_1$-minimization presented in [137] to also include additional side constraints. We present a very general null space property, which characterizes uniform recovery of all sufficiently sparse elements, under some technical assumptions. These assumptions encompass crucial properties which need to be satisfied in a specific setting in order to allow for successful sparse recovery, at least when using the setup presented in this thesis. Moreover, we show that a minor strengthening of the presented null space property allows for stable and robust recovery. Lastly, we shortly mention the case of individual recovery. Throughout this chapter, we derive four settings which are well-known in the literature from our proposed framework, namely sparse vectors, sparse nonnegative vectors, low-rank matrices and low-rank positive semidefinite matrices. We show that the results we obtain throughout that chapter simplify to the known results from the literature. These settings serve as running examples for illustration.

In order to show the generality of the framework, we treat three specific settings in more detail in Chapter 3. First, we consider a block-structure on matrices, which generalizes block-structured vectors. Afterwards, we investigate integral vectors. Lastly, we turn our attention towards so-called *constant modulus* constraints. This constraint on a complex vector demands that the absolute value of each entry is either 0 or 1. All these three settings are derived from the framework in Chapter 2, and the resulting recovery conditions in form of null space properties are presented. For block-structured matrices and vectors, we discuss the strength of the null space properties with and without an additional positive semidefiniteness or nonnegativity constraint. In particular, we construct an infinite family of instances which satisfies the NSP for block-sparse nonnegative vectors, but violates both the NSP for block-sparse vectors and for sparse nonnegative vectors. For constant modulus constraints we further present and evaluate a specific algorithm for solving the resulting recovery problem, which takes the specific problem structure into account.

In Chapter 4, we turn our attention towards random measurement matrices. Analyzing the number of measurements needed for uniform recovery with and without exploiting the side constraints is one possibility to quantify the impact of side constraints on the recovery. In case of (Gaussian) random measurement matrices, such an analysis has already been conducted for sparse vectors with and without an additional nonnegativity constraint. While the results for sparse vectors are non-asymptotic, those for sparse nonnegative vectors only hold asymptotically for large dimension and fixed ratio of dimension and number of measurements as well as number of measurements and sparsity. We extend the explicit analysis for sparse vectors to the case of sparse nonnegative vectors in Section 4.2. As main result, we obtain a non-asymptotic bound for the minimal number of measurements needed for uniform recovery of all sufficiently sparse nonnegative vectors. Furthermore, we provide numerical and empirical evidence that exploiting the nonnegativity allows for recovery with fewer measurements. Analogously, we derive a bound for the minimal number of measurements for the case of block-structured matrices in Section 4.3, which has not been analyzed under random measurements in the literature before.

As a last step in the analysis of side constraints, Chapter 5 treats the question of how to check recovery conditions. Section 5.1 provides a mixed-integer programming (MIP) formulation to verify whether a given measurement matrix satisfies the NSP for uniform recovery of sparse and sparse nonnegative vectors as well as block-structured and block-structured nonnegative vectors, respectively. A short numerical evaluation of the performance of these formulations shows that the null space property for sparse nonnegative vectors is easier to verify compared to the null space property for sparse vectors. Additionally, using these formulations, we again substantiate the result from Chapter 4 by empirically demonstrating that a (Gaussian) random measurement matrix satisfies the NSP for sparse nonnegative vectors for fewer number of measurements than needed for satisfying the NSP for sparse vectors. Afterwards, we shift our attention to the RIP and introduce its known formulation as MISDP in Section 5.2. For this formulation, we discuss several components which can be exploited in the solution process in Section 5.3. A numerical evaluation follows in Section 6.6.4, in the subsequent chapter.

Having the MISDP formulation of the RIP in mind, we consider presolving methods for MISDPs in Chapter 6. Presolving in general means to transform an instance of an optimization problem into an equivalent one by exploiting structure in the instance. The goal is to obtain an instance which is easier to solve. In contrast to MIPs, presolving for MISDPs has not yet been treated extensively in the literature. After giving a brief introduction to the solution approaches for MISDPs and existing presolving techniques in Section 6.1, we introduce several new methods for presolving and propagation in MISDPs in Sections 6.2 to 6.5. The main contribution of this chapter is a numerical evaluation of the presented presolving techniques for several classes of MISDPs in Section 6.6. After this general evaluation, we focus on the RIP and compare the effect of the presolving methods and the specialized methods introduced in Section 5.3 in more detail.

Chapter 7 concludes the obtained results and mentions several remaining open questions as well as natural extensions and problems building on the results which have not been treated throughout this thesis.

Some of the work presented in this thesis are based on results from different preprints and publications. Sections 2.1 to 2.2 as well as Section 3.1 have appeared in joint work with Janin Heuer, Thorsten Theobald and Marc E. Pfetsch [128]. Moreover, the application and the solution algorithm in Section 3.3 appeared in [97], which is joint work with Tobias Fischer, Ganapati Hegde, Marius Pesavento and Marc E. Pfetsch. Lastly, Chapter 6 appeared in similar form in the preprint [174], which is submitted for publication and is again joint work with Marc E. Pfetsch.

**Remark 1.1.** Within “EXPRESS”, the author was also involved in work on direction finding in linear arrays [175], which is not mentioned in this thesis. Direction finding means to estimate the directions from which a set of signals impinge on sensors which form a linear array. Given a signal which is transmitted by the sources, the output at the sensors can be computed using a so-called *steering matrix.* The used array geometry, i.e., the relative positions of the sensors within the array, implies that it may not be possible to uniquely identify every possible set of directions, which leads to so-called *ambiguous* directions, which cannot be differentiated by the linear array. Knowing such ambiguities beforehand is therefore of importance when designing the sensor array. In [175], a mixed-integer programming formulation is presented to compute a subset of all ambiguities a linear array geometry suffers from. This formulation emerges by making use of the underlying combinatorial structure within the steering matrix and a relation to roots of unity which sum to zero.

## 1.3 Notation and Preliminaries

In the following, we introduce notation that will be used frequently throughout this thesis. We assume that the reader has basic knowledge of linear algebra, probability theory and optimization. A good source for background in linear algebra and matrix analysis is Horn and Johnson [130], whereas for probability theory we refer to Ross [213] and basics in (convex and integer) optimization can be obtained from Boyd and Vandenberghe [31] and Schrijver [218].

For an integer $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\} \subset \mathbb{Z}^n$. The *cardinality* of a set $S \subseteq \mathbb{R}$ is denoted by $|S|$, and $\overline{S} = \mathbb{R} \setminus S$ denotes the *complement* of $S$. For a vector $x \in \mathbb{R}^n$, we frequently use the *$\ell_q$-norms* $\|x\|_q := (\sum_{i=1}^n |x_i|^q)^{1/q}$, with $1 \leq q < \infty$, as well as the $\ell_\infty$-norm $\|x\|_\infty := \max\{|x_i| \,:\, x_i \in [n]\}$. The *$\ell_q$-quasinorms* are also defined for $0 < q < 1$. Furthermore, the *support* of a vector is denoted by $\operatorname{supp}(x)$, and we define $\|x\|_0 := |\operatorname{supp}(x)|$. Note that $\|\cdot\|_0$ satisfies positive definiteness and the triangle inequality, but it is no norm, since it lacks homogeneity. Nevertheless, it is commonly referred to as $\ell_0$-norm in the literature, and we will use this name as well.

Let $S \subseteq [n]$ be a subset of indices and let $x \in \mathbb{R}^n$ be a vector. Then, $x_S$ can denote either the restriction of $x$ to indices in $S$, so that $x \in \mathbb{R}^S$, or $x_S \in \mathbb{R}^n$ as well with $(x_S)_i = 0$ for all $i \notin S$ and $(x_S)_i = x_i$ for all $i \in S$. We will use the notation $x_S$ for both cases, and depending on the context it should always be clear, which definition applies. For two vectors $x, y \in \mathbb{R}^n$, the term $\langle x, y \rangle = x^\top y$ denotes the usual standard (Euclidean) inner product on $\mathbb{R}^n$, where $x^\top$ is the transpose of $x$.

Let $A \in \mathbb{R}^{m \times n}$ be a real $m \times n$ matrix. Its *null space* is defined as the set

$$\operatorname{null}(A) := \{v \in \mathbb{R}^n \,:\, Av = 0\}.$$

The *transpose* of $A$ is denoted by $A^\top$. We frequently use the *Frobenius norm* $\|A\|_F$ and the *nuclear norm* $\|A\|_*$, which are defined as

$$\|A\|_F := \Big(\sum_{i=1}^{m}\sum_{j=1}^{n}|A_{ij}|^2\Big)^{\frac{1}{2}} = \Big(\sum_{k=1}^{\min\{m,n\}} \sigma_k(A)^2\Big)^{\frac{1}{2}},$$

$$\|A\|_* := \sum_{k=1}^{r} \sigma_k(A),$$

respectively, where $\sigma_1(A), \ldots, \sigma_r(A)$ are the *singular values* of $A$. Consequently, the nuclear norm is the $\ell_1$-norm of the vector of singular values. For $A$, $B \in \mathbb{R}^{m \times n}$, we use the *(Frobenius) inner product*

$$\langle A, B\rangle_{\mathrm{F}} := \operatorname{tr}\left(A^\top B\right) = \sum_{i,j=1}^{n} A_{ij}\, B_{ij}, \tag{1.1}$$

where $\operatorname{tr}(X) := \sum_{i=1}^{n} X_{ii}$ is the *trace* of the square matrix $X \in \mathbb{R}^{n\times n}$.

We denote the space of $n\times n$ real symmetric matrices with $\mathcal{S}^n$. The matrix $A \in \mathcal{S}^n$ is called *positive semidefinite (psd)*, denoted $A \succeq 0$, if $x^\top Ax \geq 0$ for all $x \in \mathbb{R}^n$, or, equivalently, if all *eigenvalues* $\lambda(A)$ of $A$ are nonnegative. The notation $A \preceq 0$ is used if $A$ is *negative semidefinite*, i.e., $-A$ is psd. The space of (symmetric) psd matrices is denoted by $\mathcal{S}^n_+$. The notations $\mathbb{1}$, $\mathbf{1}$ and $\mathbb{I}$ are used for the *all-ones vector*, the *all-ones matrix* and the *identity matrix* with dimension depending on the context, whereas the all-zeros vector and matrix are simply denoted by 0. Finally, $\operatorname{Diag}(x)$ denotes a *diagonal matrix* containing the vector $x$ along its diagonal.

When comparing computational results, we make use of the *arithmetic*, *geometric* and *shifted geometric mean.* For values $v_1, \dots, v_n$, the arithmetic mean is defined as $\frac{1}{n}(\sum_{i=1}^{n} v_i)$, the geometric mean is computed according to $(\prod_{i=1}^{n} v_i)^{1/n}$ and the shifted geometric mean additionally applies a shift $s$ to the values and computes the geometric mean of the shifted values. Afterwards, the initial shift is subtracted again from the result. Thus, the shifted geometric mean is defined as

$$\left(\prod_{i=1}^{n}(v_i + s)\right)^{1/n} - s. \tag{1.2}$$

Compared to the arithmetic mean, the shifted geometric mean is more robust against very large outliers, and compared to the geometric mean, the shifted geometric mean is also more robust against very small outliers, see Achterberg [1]. We use a shift of $s = 1$ second for solution times and a shift of $s = 100$ for the number of nodes processed within a branch-and-bound approach, which is one prominent method to solve MIPs. Branch-and-bound was first proposed by Land and Doig [153] in 1960 and extended to general nonlinear problems by Dakin [58]. It consists of dividing the set of feasible solutions into smaller subsets by creating subproblems. This is achieved by adding variable bounds or additional constraints. For each subproblem, a continuous relaxation is solved in order to obtain lower bounds. If a feasible solution to the original problem is found, this yields an upper bound. If the lower bound of a subproblem is larger than the upper bound, i.e., the objective value of the currently best feasible solution, this subproblem can be disregarded and the corresponding node can be pruned. In order to improve the lower bounds, the subproblems can be strengthened, e.g., by adding (linear) inequalities which are valid for feasible integral solutions but cut off fractional solutions of the relaxation.

CHAPTER 2

# A General Framework for Recovery Using Null Space Properties

The classical sparse recovery problem asks to recover a sparse vector $x^{(0)} \in \mathbb{R}^n$ from its measurements $Ax^{(0)}$, where $A \in \mathbb{R}^{m\times n}$ is a measurement matrix. Using the $\ell_0$-norm to search for the sparsest vector $x \in \mathbb{R}^n$ with $Ax = Ax^{(0)}$ leads to $(P_0)$, that is, $\min\{\|x\|_0 \,:\, Ax = Ax^{(0)}\}$, which is $\mathcal{NP}$-hard [185]. A convex approximation can be obtained by replacing the $\ell_0$-norm with the $\ell_1$-norm, which leads to the convex optimization problem $(P_1)$, that is, $\min\{\|x\|_1 \,:\, Ax = Ax^{(0)}\}$. Uniform recovery is characterized by the classical NSP

$$\|v_S\|_1 < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\},\ \forall\, S \subseteq [n],\ |S| \leq s, \tag{NSP}$$

see, e.g., Foucart and Rauhut [104, Theorem 4.5] and Cohen et al. [55]. If (NSP) is satisfied for the matrix $A$, then every sufficiently sparse $x^{(0)} \in \mathbb{R}^n$ is the unique optimal solution of $(P_1)$.

The sparse recovery problem for vectors can be extended to matrices as well, since sparsity translates to low-rankness. Recall that the rank of a matrix $X \in \mathbb{C}^{m\times n}$ is the $\ell_0$-norm of the vector $\sigma(X)$ of singular values of $X$. In order to recover a symmetric low-rank matrix $X^{(0)} \in \mathcal{S}^n$ from its measurements $\mathcal{A}(X^{(0)})$, where $\mathcal{A}\colon \mathcal{S}^n \mapsto \mathbb{R}^m$ is a linear sensing operator, Fazel [94] suggested to solve the optimization problem

$$\min\{\operatorname{rank}(X) \,:\, \mathcal{A}(X) = \mathcal{A}(X^{(0)}),\ X \in \mathcal{S}^n\}. \tag{2.1}$$

Similar to $\|\cdot\|_0$, the rank is a nonconvex function, so that (2.1) is hard to solve in practice. The analog of the $\ell_1$-norm as convex relaxation of $\|\cdot\|_0$ for vectors is the nuclear norm $\|X\|_*$, i.e., the $\ell_1$-norm of the vector $\sigma(X)$ of singular values of $X$. This leads to the convex optimization problem

$$\min\{\|X\|_* : A(X) = A(X^{(0)}),\ X \in \mathcal{S}^n\}. \tag{2.2}$$

For more information to low-rank matrix recovery, see, e.g., Recht et al. [210] and the references therein.

Additional knowledge, e.g., if the original sparse vector $x^{(0)}$ is known to be nonnegative, or if the original low-rank matrix $X^{(0)}$ is known to be positive semidefinite, can be included in the recovery problems as well. More precisely, in these cases, the additional side constraints $x \geq 0$ or $X \succeq 0$ can be added to the recovery problems $(P_1)$ and (2.2), respectively. For all these settings, adaptions of (NSP) and other recovery conditions for individual and uniform recovery are known in the literature. In the classical case, the corresponding NSP can be found in Gribonval and Nielsen [123] and in [104, Theorem 4.4]. If the vectors have to be nonnegative, the respective NSP appears in Khajehnejad et al. [143] and in Zhang [256]. For the case of arbitrary matrices or positive semidefinite (psd) matrices, corresponding NSPs can be found in Kong et al. [146], Oymak and Hassibi [192], or in [104, Theorem 4.40]. NSPs for block-sparse vectors have first been considered in Stojnic et al. [230], and NSPs for block-sparse nonnegative vectors appear in Stojnic [229]. In [128], NSPs for block-sparse as well as block-sparse positive semidefinite matrices were derived. The general framework presented by Juditsky et al. [137] subsumes many existing NSPs, but additional side constraints such as nonnegativity or positive semidefiniteness cannot be included.

In this chapter, we extend the framework by Juditsky et al. [137] by also incorporating additional side constraints such as nonnegativity, positive semidefiniteness or integrality. This yields a general framework for individual and uniform recovery under side constraints using null space properties. The main result of this chapter is a general null space condition which characterizes exact recovery of sufficiently sparse vectors from their measurements, i.e., uniform recovery, using an appropriate general recovery problem. Section 2.1 introduces the framework and formulates necessary technical assumptions. In Section 2.2 a general NSP is presented and proved to characterize uniform recovery in the framework. The subsequent Sections 2.3 and 2.4 extend this framework further to also cover stable and robust recovery, as well as recovery of a fixed sufficiently sparse vector, i.e., individual recovery.

The general framework and the null space property presented in Sections 2.1 and 2.2 have been published in [128], which is joint work with Janin Heuer, Thorsten Theobald and Marc E. Pfetsch. For this thesis, we added a short discussion about the connections to the concept of decomposable norms. Moreover, we extend the framework to also cover stable and robust recovery in Section 2.3. In doing so, it turned out that a minor adaption is needed in order to also cover stability and robustness. This adaption implies that only one null space property is needed in comparison to the pair of null space properties proposed in [128]. The details of this adaption and its implications are discussed in Remarks 2.4, 2.9 and 2.18.

## 2.1 Components of the General Framework

As in the framework by Juditsky et al. [137], let $\mathcal{X}$ and $\mathcal{E}$ be two Euclidean spaces. The signal space $\mathcal{X}$ is the space in which the signals of interest lie, and the space $\mathcal{E}$ models representations of the signals. We use a *linear sensing map* $A\colon \mathcal{X} \to \mathbb{R}^m$ to acquire signals $x \in \mathcal{X}$ and a *linear representation map* $B\colon \mathcal{X} \to \mathcal{E}$ to map a signal to an appropriate representation. In this chapter, unless otherwise stated, the image of $x$ under a linear operator $F$ is denoted by $Fx$.

**Remark 2.1.** Throughout this chapter, we consider real vector spaces $\mathcal{X}$ and $\mathcal{E}$, since this is the more natural setting when considering nonnegative vectors. However, at least for unrestricted (block-)vectors or (block-diagonal) matrices, the null space properties and statements in the subsequent Sections 2.2 to 2.4 also carry over without changes to the situation where the spaces $\mathcal{X}$ and $\mathcal{E}$ are complex spaces.

The classical setting of sparse recovery, where $x$ is sparse in its “natural” representation, can be obtained by choosing $\mathcal{X} = \mathcal{E}$ and $B$ to be the identity. However, the framework also covers the so-called “analysis” setting, where the signal $x$ is only sparse in a suitable representation system, with $B$ being an appropriate transformation; see, e.g., for an overview [47, 87, 138, 184]. Examples for transformations include the discrete Fourier transform, different wavelet transforms [42, 124, 171, 212, 221] or a finite difference operator in total variation minimization [36, 48, 187].

In order to model additional side constraints such as nonnegativity, we introduce a set $\mathcal{C} \subset \mathcal{X}$ with $0 \in \mathcal{C}$ and its image $\mathcal{D} := \{Bx : x \in \mathcal{C}\} \subseteq \mathcal{E}$ under the map $B$. Additionally, let $\|\cdot\|$ be a norm on $\mathcal{E}$. The framework uses projections onto appropriate subspaces to express sparsity of elements $x \in \mathcal{X}$. Therefore, consider a set $\mathcal{P}$ of linear maps on $\mathcal{E}$ and a map $\nu\colon \mathcal{P} \to \mathbb{R}_+$. Each map $P \in \mathcal{P}$ is assigned a nonnegative real weight $\nu(P)$ and another linear map $\overline{P}\colon \mathcal{E} \to \mathcal{E}$. A general concept of sparsity can now be defined as follows.

**Definition 2.2** (Sparsity). *Let $s \in \mathbb{R}_+$. An element $y \in \mathcal{E}$ is called $s$-*sparse *if there exists a linear map $P \in \mathcal{P}$ with $\nu(P) \leq s$ and $Py = y$. Accordingly, $x \in \mathcal{X}$ is* *called* $s$-sparse*, if its representation $Bx \in \mathcal{E}$ is $s$-sparse, i.e., if there exists $P \in \mathcal{P}$ with $\nu(P) \leq s$ and $PBx = Bx$. Furthermore, let $\mathcal{P}_s \coloneqq \{P \in \mathcal{P} : \nu(P) \leq s\}$ be the set of linear maps that can allow $s$-sparse elements.*

We do not assume that $\nu(P)$ is integer-valued in general, even if this is the case in most examples of this general framework.

For a given right-hand side $b \in \mathbb{R}^m$, the corresponding generalized recovery problem can be formulated as

$$\min \{\|Bx\| \,:\, Ax = b,\ x \in \mathcal{C}\}, \tag{2.3}$$

which is a convex optimization problem, if $\mathcal{C}$ is convex. We assume that $\mathcal{C}$ is closed, such that the minimum in (2.3) is attained and finite, or $\infty$.

If $\mathcal{C} = \mathcal{X}$, then our framework reduces to the framework by Juditsky et al. [137], and our statements become the statements concerning noise-free recovery in [137], see Remark 2.11 below.

In order to get an intuition, the following examples derive many important cases previously considered in the literature as special cases of the framework described above. These cases are used as running examples throughout the chapter to demonstrate the obtained results. For a (finite) set $I$, let $\mathcal{E}_I \coloneqq \{y \in \mathcal{E} : y_i = 0\ \forall\, i \notin I\}$ be its coordinate subspace and let $\mathbb{R}^I$ denote the space of elements with real entries indexed by the elements of $I$.

**Example 2.3.**

(2.3.1) *Recovery of sparse vectors by $\ell_1$-minimization*

For the recovery of sparse vectors $x \in \mathbb{R}^n$, let $\mathcal{X} = \mathcal{E} = \mathcal{C} = \mathbb{R}^n$, $B$ be the identity and $\|\cdot\| = \|\cdot\|_1$, so that $\mathcal{D} = \mathbb{R}^n$. Let $\mathcal{P}$ be the set of orthogonal projectors onto all coordinate subspaces of $\mathbb{R}^n$, and define $\overline{P} \coloneqq I_n - P$, where $I_n$ denotes the identity mapping on $\mathbb{R}^n$. If $P$ projects onto $\mathcal{E}_S$ for an index set $S \subseteq [n]$, then $PBx = Px = x_S$ and $\overline{P}Bx = x_{\overline{S}}$, where $\overline{S} \coloneqq [n] \setminus S$ is the complement of $S$. Define the nonnegative weight $\nu(P) \coloneqq \operatorname{rank}(P)$, i.e., $\nu(P) = \dim(\mathcal{E}_S) = |S|$. In this case, the notion of general sparsity coincides with the classical sparsity of nonzero entries in a vector $x \in \mathbb{R}^n$ (if $Px = x$, then $\|x\|_0 \leq \nu(P)$), and the recovery problem (2.3) becomes classical $\ell_1$-minimization.

(2.3.2) *Recovery of sparse nonnegative vectors by $\ell_1$-minimization*

For the recovery of nonnegative vectors let $\mathcal{X}$, $\mathcal{E}$, $B$, $\mathcal{P}$, $\nu(P)$, $\overline{P}$, $\|\cdot\|$ be defined as in the previous example, and let $\mathcal{C} = \mathbb{R}^n_+$, so that $\mathcal{D} = \mathbb{R}^n_+$. As before, we have $PBx = x_S$ and $\overline{P}Bx = x_{\overline{S}}$ and the notion of general sparsity simplifies to the classical sparsity of nonzero entries in a nonnegative vector $x \in \mathbb{R}^n_+$. Consequently, the recovery problem (2.3) becomes nonnegative $\ell_1$-minimization.

(2.3.3) *Recovery of low-rank matrices by nuclear norm minimization*
Let $\mathcal{X} = \mathcal{E} = \mathcal{C} = \mathbb{R}^{n_1 \times n_2}$. Let the representation map $B$ be the identity (so that $\mathcal{D} = \mathbb{R}^{n_1 \times n_2}$), and let the norm $\|\cdot\|$ be the nuclear norm $\|\cdot\|_*$. For some positive integer $k$ and a set $I \subseteq [k]$, define the matrix $T_I^k \in \mathbb{R}^{k \times k}$ to be a matrix with ones on the diagonal at positions $(i,i)$ for $i \in I$ and zeros elsewhere. Let $\mathcal{O}^k$ be the set of $k \times k$ orthogonal matrices. Then define the set $\mathcal{P}$ of projections $P \colon \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ as

$$\mathcal{P} := \left\{X \mapsto U\, T_I^{n_1}\, U^\top\, X\, V\, T_I^{n_2}\, V^\top \;:\; U \in \mathcal{O}^{n_1},\, V \in \mathcal{O}^{n_2},\, I \subseteq [n_{\min}]\right\},$$

where $n_{\min} = \min\{n_1, n_2\}$. For $P \in \mathcal{P}$ defined by $U \in \mathcal{O}^{n_1}$, $V \in \mathcal{O}^{n_2}$ and an index set $I$, define $\nu(P) = |I|$, and $\overline{P}$ to be the map

$$X \mapsto U\,(\mathbb{I}_{n_1} - T_I^{n_1})\,U^\top\, X\, V\,(\mathbb{I}_{n_2} - T_I^{n_2})\,V^\top,$$

where $\mathbb{I}_{n_i}$ denotes the identity matrix of size $n_i$, so that $\mathbb{I}_{n_i} - T_I^{n_i} = T_{[n_i]\setminus I}^{n_i}$.[1]
The intuition behind these projections is as follows. If $U$ and $V$ are chosen such that $X = U\Sigma V^\top$ is the singular value decomposition of $X$, then $P$ first projects $X$ onto $\Sigma$ containing the singular values $\sigma_1(X) \geq \cdots \geq \sigma_{\min\{n_1,n_2\}}(X)$, then sets $\sigma_i(X) = 0$ for all $i \notin I$ via left- and right-multiplication of $T_I^{n_1}$ and $T_I^{n_2}$, respectively, and transforms the resulting diagonal matrix $\tilde{\Sigma}$ back by $U\tilde{\Sigma}V^\top$.

A matrix $X \in \mathbb{R}^{n_1 \times n_2}$ is rank-$s$-sparse, i.e., there exist at most $s$ nonzero singular values, if and only if there exists a projection $P \in \mathcal{P}$ with corresponding index set $I$ with $PX = X$ and $|I| \leq s$. This implies $\sigma_i(X) = 0$ for all $i \notin I$, so that $\operatorname{rank}(X) \leq s$. Therefore, the recovery problem (2.3) becomes low-rank matrix recovery, and sparsity translates to low-rankness.

(2.3.4) *Recovery of low-rank positive semidefinite matrices by nuclear norm minimization*
For the recovery of positive semidefinite matrices, consider $\mathcal{X} = \mathcal{E} = \mathcal{S}^n$, $\mathcal{C} = \mathcal{S}_+^n$, and let $B$ be the identity map (thus, $\mathcal{D} = \mathcal{S}_+^n$). The definitions of $\mathcal{P}$, $\overline{P}$, $\nu(P)$ and $\|\cdot\|$ are as in the previous example. Again, the notion of sparsity simplifies to low-rankness, and the recovery problem (2.3) becomes low-rank matrix recovery for positive semidefinite matrices.

Nontrivial representation maps $B \colon \mathcal{X} \to \mathcal{E}$ appear for example in the case of settings in which vectors or matrices obey a certain block-structure or block-diagonal form, even with overlapping blocks. These settings will be discussed in more detail in the next chapter.

[1] Note that $U\, T_I^{n_1}\, U^\top\, X\, V\, T_I^{n_2}\, V^\top$ and $U\,(\mathbb{I}_{n_1} - T_I^{n_1})\,U^\top\, X\, V\,(\mathbb{I}_{n_2} - T_I^{n_2})\,V^\top$ denote matrix products, since $U$, $T_I^{n_1}$, $T_I^{n_2}$ $X$, and $V$ are matrices, and not linear maps.

The next section states some technical assumptions which are needed to characterize recovery of sparse elements $x^{(0)}$ from its measurements $b = Ax^{(0)}$ using the recovery problem (2.3).

## Assumptions

We consider the following assumptions on the sets $\mathcal{C}$, $\mathcal{D}$, $\mathcal{P}$ and the norm $\|\cdot\|$.

(A1) For every $P \in \mathcal{P}$, it holds that

- $P^2 = P$, i.e., $P$ is a projector, and
- $Py \in \mathcal{D}$ for all $y \in \mathcal{D}$.

Moreover, $B\colon \mathcal{X} \to \mathcal{E}$ is injective, and for all $c_1$, $c_2 \in \mathcal{C}$, $c_1 + c_2 \in \mathcal{C}$ holds.

(A2) For every $P \in \mathcal{P}$, the corresponding linear map $\overline{P}\colon \mathcal{E} \to \mathcal{E}$ satisfies

- $\overline{P}P = 0$, and
- $\overline{P}y \in \mathcal{D}$ for all $y \in \mathcal{D}$.

(A3) For all $y \in \mathcal{E}$ and all $P \in \mathcal{P}$, it holds that $y = Py + \overline{P}y$.

(A4) For all $s \geq 0$, $P \in \mathcal{P}_s$, for all $x$, $z \in \mathcal{C}$ and $v := x - z$ and all $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ so that $\|B\hat{v}^{(2)}\|$ is minimal among all such decompositions it holds that

$$\|Bx\| \leq \|Bz\| + \|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| - \|\overline{P}Bv\| + 2\|\overline{P}Bx\|. \tag{2.4}$$

For $v \in \mathcal{C} + (-\mathcal{C})$, a decomposition $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $\|B\hat{v}^{(2)}\|$ minimal as used in Assumption (A4) is also called *minimal decomposition of* $v$. Moreover, the minimum

$$\min \{\|B\hat{v}^{(2)}\| \,:\, v = \hat{v}^{(1)} - \hat{v}^{(2)},\ \hat{v}^{(1)},\ \hat{v}^{(2)} \in \mathcal{C}\}$$

is attained, since we assumed that $\mathcal{C}$ is closed.

**Remark 2.4.** Let us shortly compare the presented assumptions with those in [128]. Assumptions (A1) to (A3) remain the same, however, in [128], two different versions of Assumption (A4) are used, namely (A4a) which demands that Inequality (2.4) holds for *all* decompositions $v = v^{(1)} - v^{(2)}$ and (A4b) which only states that there *exists* a decomposition which satisfies the inequality. This leads to two different versions of a null space property, depending on which assumption is satisfied. A closer inspection reveals that under the stronger Assumption (A4a), both null space properties are in fact equivalent, whereas under Assumption (A4b), this is wrong in general. In order to extend the proposed framework to also cover stable and robust recovery, it turns out that Assumption (A4b) needs to be modified to demand that the inequality holds for *all minimal* decompositions. This implies that one of the two null space properties needs to be adapted as well. However, after this modification, the two null space properties are equivalent, regardless whether (A4a) or (A4b) holds. Thus, in order to have a simplified exposition, we only use the modified version of Assumption (A4b) as Assumption (A4) and also only the modified null space property, see also Remarks 2.9 and 2.18.

**Remark 2.5.** Note that Assumption (A4) is asymmetric. Indeed, if we require $\|Bv^{(1)}\|$ to be minimal instead of $\|Bv^{(2)}\|$, then Assumption (A4) is violated even for the classical case of recovery of sparse vectors using the $\ell_1$-norm, as the following simple example shows. Consider $\mathcal{C} = \mathcal{X} = \mathbb{R}^n$ with $n \geq 7$ and the vectors

$$x_S = (-4,\, 5,\, -6)^\top,\; z_S = (1,\, 2,\, -3)^\top,\; v_S^{(1)} = (0,\, 0,\, 0)^\top,\; v_S^{(2)} = (5,\, -3,\, 3)^\top,$$

with $x_{\overline{S}} = z_{\overline{S}} = v_{\overline{S}}^{(1)} = v_{\overline{S}}^{(2)} = 0$ for some index set $S$ and $v = x - z = v^{(1)} - v^{(2)}$. Then,

$$\|z\|_1 - \|x\|_1 + \|v_S^{(1)}\|_1 - \|v_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1 = -20 < 0.$$

With $v_S^{(1)} = (-5,\, 3,\, -3)^\top$, $v_S^{(2)} = (0,\, 0,\, 0)^\top$, i.e., $\|v^{(2)}\|_1$ minimal, we obtain

$$\|z\|_1 - \|x\|_1 + \|v_S^{(1)}\|_1 - \|v_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1 = 2 > 0.$$

This suggests that the asymmetry in Assumption (A4) is necessary and cannot be replaced by a symmetric condition.

The different settings in Example 2.3 satisfy Assumptions (A1) to (A3), since $\mathcal{P}$ consists of orthogonal projections. Only Assumption (A4) remains to be verified. This will be discussed after the main result of uniform recovery in the next section. But first we shortly compare the framework introduced above with the concept of decomposable norms which has first been considered by Negahban et al. [188] and is used by Candès and Recht [44] and Roulet et al. [214] to present unified bounds for recovery.

**Decomposable norms** One simple, but important, common property of the $\ell_1$-norm, the nuclear norm and the mixed $\ell_2/\ell_1$-norm from Example 2.3 is the so-called *decomposability.* The concept of decomposable norms has first been defined in [188] using subspaces. An equivalent definition in terms of projectors is given in [214], which we recall here.

**Definition 2.6** (Roulet et al. [214]). *Let $\mathcal{E}$ be an Euclidean space and $\|\cdot\|$ be a norm on $\mathcal{E}$. For a set $\mathcal{P}$ of orthogonal projectors, the norm $\|\cdot\|$ is called* decomposable *with respect to $\mathcal{P}$, if the following properties are satisfied.*

*(i) Each $P \in \mathcal{P}$ is assigned an orthogonal projector $\overline{P}$ with $P\overline{P} = \overline{P}P = 0$ and a nonnegative weight $\nu(P)$.*

*(ii) $\|Px + \overline{P}x\| = \|Px\| + \|\overline{P}x\|$ holds for all $x \in \mathcal{E}$ and all $P \in \mathcal{P}$.*

Note that Part (i) of Definition 2.6 is contained in Assumptions (A1) and (A2). The subsequent lemma compares Assumptions (A3) and (A4) and Part (ii) of Definition 2.6.

**Lemma 2.7.** *Let Assumptions (A1) and (A2) hold. Then, Assumption (A4) implies*

$$\|Bz\| = \|PBz\| + \|\overline{P}Bz\| \tag{DP}$$

*for all $s \geq 0$, $P \in \mathcal{P}_s$ and all $z \in \mathcal{C}$. In case that $\mathcal{C} = \mathcal{X}$, the converse also holds, i.e., Assumption (A4) and* (DP) *are equivalent.*

*If additionally Assumption (A3) holds, then* (DP) *can be written as*

$$\|PBz + \overline{P}Bz\| = \|PBz\| + \|\overline{P}Bz\|.$$

*Proof.* Let Assumptions (A1), (A2) and (A4) hold. Let $s \geq 0$, $P \in \mathcal{P}_s$ as well as $z \in \mathcal{C}$. Then, $Bz \in \mathcal{D}$ and consequently, $PBz \in \mathcal{D}$ as well (Assumption (A1)). Thus, there exists $x \in \mathcal{C}$ with $Bx = PBz$. Define $v := x - z$. Since $P^2 = P$ (Assumption (A1)), we have $PBx = PBz = Bx$ and thus $PBv = PBx - PBz = 0$. By Assumption (A2) we have $\overline{P}Bx = \overline{P}PBz = 0$. This implies $\overline{P}Bv = -\overline{P}Bz$. Using Assumption (A4) for $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ and $\|B\hat{v}^{(2)}\|$ minimal yields

$$\begin{aligned}
0 &\leq \|Bz\| - \|Bx\| + \|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| - \|\overline{P}Bv\| + 2\|\overline{P}Bx\| \\
&= \|Bz\| - \|PBz\| + \|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| - \|-\overline{P}Bz\| \\
&\leq \|Bz\| - \|PBz\| + \|PBv\| - \|\overline{P}Bz\| \\
&= \|Bz\| - \big(\|PBz\| + \|\overline{P}Bz\|\big).
\end{aligned}$$

Combined with the triangle inequality, this shows $\|Bz\| = \|PBz\| + \|\overline{P}Bz\|$.

Now assume that additionally $\mathcal{C} = \mathcal{X}$. Then, for $v \in \mathcal{C} + (-\mathcal{C})$, its minimal decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ is unique and given by $\hat{v}^{(1)} = v$, and $\hat{v}^{(2)} = 0$. In order to show the reverse direction in this case, assume that $\|Bz\| = \|PBz\| + \|\overline{P}Bz\|$ holds for all $s \geq 0$, $P \in \mathcal{P}_s$ and all $z \in \mathcal{C}$. Let $s \geq 0$, $P \in \mathcal{P}_s$, and $x$, $z \in \mathcal{C}$ with $v := x - z$. This implies

$$\begin{aligned}
&\|Bz\| - \|Bx\| + \|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| - \|\overline{P}Bv\| + 2\|\overline{P}Bx\| \\
=& \|Bz\| - \|Bx\| + \|PBv\| - \|\overline{P}Bv\| + 2\|\overline{P}Bx\| \\
\geq& \|Bz\| - \|Bx\| + \|PBx\| - \|PBz\| - \|\overline{P}Bx\| - \|\overline{P}Bz\| + 2\|\overline{P}Bx\| \\
=& \|Bz\| - \big(\|PBz\| + \|\overline{P}Bz\|\big) - \|Bx\| + \big(\|PBx\| + \|\overline{P}Bx\|\big) \\
=& 0,
\end{aligned}$$

where we used the decomposition property $\|Bz\| = \|PBz\| + \|\overline{P}Bz\|$ twice for the last equality. This shows the desired equivalence if $\mathcal{C} = \mathcal{X}$ and finishes the proof, since Assumption (A3) implies $Bz = PBz + \overline{P}Bz$. □

Since for $\mathcal{C} = \mathcal{X}$, Assumption (A4) is equivalent to the decomposability property, the general framework presented in this section extends the concept of decomposable norms to the case where additional side constraints are present, i.e., $\mathcal{C} \neq \mathcal{X}$.

The next section presents the null space property building on the Assumptions (A1) to (A4), which leads to the main result for exact uniform recovery.

## 2.2 Uniform Recovery in the General Framework

In order to characterize uniform recovery, we define the following null space property. Let $\operatorname{null}(A) := \{v \in \mathcal{X} : Av = 0\}$ denote the null space of the linear sensing map $A$,

**Definition 2.8.** *The linear sensing map $A$ satisfies the* general null space property *of order $s$ for the set $\mathcal{C}$ if and only if for all $v \in (\operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C})))$ with $Bv \neq 0$ and all $P \in \mathcal{P}_s$ it holds that*

$$\begin{aligned}
-\overline{P}Bv \in \mathcal{D} \implies \forall\, \hat{v}^{(1)}, \hat{v}^{(2)} \in \mathcal{C} \text{ with } v = \hat{v}^{(1)} - \hat{v}^{(2)} \text{ and } \|B\hat{v}^{(2)}\| \text{ minimal:}& \\
\|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| < \|\overline{P}Bv\|.& \qquad (\mathrm{NSP}^{\mathcal{C}})
\end{aligned}$$

**Remark 2.9.** Note that this null space property corresponds to the first null space property NSP-I$^{\mathcal{C}}$ in [128]. As outlined in Remark 2.4, the extension to stable and robust recovery required a modification of one of the assumptions and consequently of NSP-I$^{\mathcal{C}}$, which becomes (NSP$^{\mathcal{C}}$). It turned out that the null space properties NSP-II$^{\mathcal{C}}$ in [128] and (NSP$^{\mathcal{C}}$) are in fact equivalent, so that only (NSP$^{\mathcal{C}}$) is used throughout this thesis.

The next result shows that uniform recovery of a sufficiently sparse $x^{(0)} \in \mathcal{C}$ from its measurements $b = Ax^{(0)}$ using (2.3) is exactly characterized by (NSP$^{\mathcal{C}}$).

**Theorem 2.10.** *Suppose that Assumptions (A1) to (A4) are satisfied. Let $A$ be a linear sensing map and $s \geq 0$. Then the following statements are equivalent:*

*(i) Every $s$-sparse $x^{(0)} \in \mathcal{C}$ is the unique solution of* (2.3) *with $b = Ax^{(0)}$.*

*(ii) $A$ satisfies the general null space property* ($\mathrm{NSP}^{\mathcal{C}}$) *of order $s$ for the set $\mathcal{C}$.*

*Proof.* In order to prove the equivalence, let $s \geq 0$ and suppose Assumptions (A1) to (A4) are satisfied.

Assume that every $s$-sparse $x^{(0)} \in \mathcal{C}$ is the unique optimal solution of the recovery problem (2.3) with $b = Ax^{(0)}$. Let $P \in \mathcal{P}_s$ and $v \in (\mathrm{null}(A) \cap (\mathcal{C} + (-\mathcal{C})))$ with $Bv \neq 0$ and $-\overline{P}Bv \in \mathcal{D}$. Let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ be a decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ so that $\|B\hat{v}^{(2)}\|$ is minimal among all such decompositions. Since $v \in \mathcal{C} + (-\mathcal{C})$, such a decomposition exists. Define

$$w_S^{(1)} := PB\hat{v}^{(1)}, \quad w_S^{(2)} := PB\hat{v}^{(2)}, \quad w_{\overline{S}} := -\overline{P}Bv.$$

There exist $v_S^{(1)}$, $v_S^{(2)} \in \mathcal{C}$ with $Bv_S^{(1)} = w_S^{(1)}$ and $Bv_S^{(2)} = w_S^{(2)}$, since $PBx \in \mathcal{D}$ for all $x \in \mathcal{C}$ by Assumption (A1). Moreover, $-\overline{P}Bv \in \mathcal{D}$ by assumption, so that there exists $v_{\overline{S}} \in \mathcal{C}$ with $Bv_{\overline{S}} = w_{\overline{S}}$. Assumption (A3) implies

$$\begin{aligned} Bv = PBv + \overline{P}Bv = PB\hat{v}^{(1)} - PB\hat{v}^{(2)} + \overline{P}Bv &= Bv_S^{(1)} - Bv_S^{(2)} - Bv_{\overline{S}} \\ &= B(v_S^{(1)} - v_S^{(2)} - v_{\overline{S}}), \end{aligned}$$

which yields $v = v_S^{(1)} - v_S^{(2)} - v_{\overline{s}}$, since $B$ is injective by Assumption (A1). Accordingly,

$$0 = Av = A(v_S^{(1)} - v_S^{(2)} - v_{\overline{s}}) \quad \Leftrightarrow \quad A(v_S^{(2)} + v_{\overline{s}}) = Av_S^{(1)}.$$

Assumption (A1) implies that $v_S^{(1)}$ is $s$-sparse, since

$$PBv_S^{(1)} = Pw_S^{(1)} = PPB\hat{v}^{(1)} = PB\hat{v}^{(1)} = w_S^{(1)} = Bv_S^{(1)}.$$

By construction, $v_S^{(1)}$, $v_S^{(2)}$, $v_{\overline{S}} \in \mathcal{C}$, so that Assumption (A1) yields $v_S^{(2)} + v_{\overline{S}} \in \mathcal{C}$ as well. Thus, the uniqueness property of $A$ for the $s$-sparse $v_S^{(1)}$ implies

$$\begin{aligned} & \|Bv_S^{(1)}\| < \|Bv_S^{(2)} + Bv_{\overline{s}}\| \leq \|Bv_S^{(2)}\| + \|Bv_{\overline{s}}\| \\ \Leftrightarrow \quad & \|Bv_S^{(1)}\| - \|Bv_S^{(2)}\| - \|Bv_{\overline{s}}\| < 0 \\ \Leftrightarrow \quad & \|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| - \|\overline{P}Bv\| < 0, \end{aligned}$$

which shows the desired general null space property ($\mathrm{NSP}^{\mathcal{C}}$) of order $s$ for the set $\mathcal{C}$.

For the reverse direction, assume $A$ satisfies the general null space property ($\mathrm{NSP}^{\mathcal{C}}$) of order $s$ for the set $\mathcal{C}$. Let $x$, $z \in \mathcal{C}$ with $Bx \neq Bz$, $Ax = Az$ and $x$ being $s$-sparse, i.e., there exists $P \in \mathcal{P}_s$ with $PBx = Bx$. In order to show the uniqueness property for $A$, we need to prove that $\|Bx\| < \|Bz\|$. Define $v := x - z \in \mathrm{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ with

$$-\overline{P}Bv = -\overline{P}Bx + \overline{P}Bz = -\overline{P}PBx + \overline{P}Bz = \overline{P}Bz \in \mathcal{D},$$

since $\overline{P}P = 0$ and $\overline{P}y \in \mathcal{D}$ for all $y \in \mathcal{D}$ by Assumption (A2). The general null space property ($\mathrm{NSP}^{\mathcal{C}}$) implies that

$$\|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| - \|\overline{P}Bv\| < 0$$

for $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$, where $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ is a decomposition such that $\|B\hat{v}^{(2)}\|$ is minimal. For this minimal decomposition, Assumption (A4) yields

$$\|Bx\| \leq \|Bz\| + \|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| - \|\overline{P}Bv\| + 2\|\overline{P}Bx\| < \|Bz\|,$$

since by assumption, $\overline{P}Bx = \overline{P}PBx = 0$. Thus, $x$ must be the unique solution of (2.3), which completes the proof. □

It is worth mentioning that many transformations $B$ which are used in the analysis setting, e.g., the finite difference operator in total variation minimization, are not injective. In this case, the null space property ($\mathrm{NSP}^{\mathcal{C}}$) is still sufficient for uniform recovery, but no longer necessary, since the injectivity of $B$ is only needed for necessity in the proof of Theorem 2.10. This is in line with the known NSPs in the analysis setting in the literature, see, e.g., Kabanava and Rauhut [138] or Krahmer et al. [147, Corollary 2.1].

**Remark 2.11.** Let $\mathcal{C} = \mathcal{X}$ and $\mathcal{D} = \mathcal{E}$. Then our setting simplifies to the framework by Juditsky et al. [137], who define a sparsity structure on $\mathcal{E}$ as a norm $\|\cdot\|$ on $\mathcal{E}$ together with a family $\mathcal{P}$ of linear maps of $\mathcal{E}$ into itself, satisfying the following assumptions:

A.1) $P^2 = P$ for all $P \in \mathcal{P}$, i.e., every $P$ is a projection;

A.2) Every $P \in \mathcal{P}$ is assigned a nonnegative weight $\nu(P)$ and a linear map $\overline{P}$ on $\mathcal{E}$ with $\overline{P}P = 0$;

A.3) For all $P \in \mathcal{P}$ and $x$, $y \in \mathcal{E}$, one has $\|P^*x + \overline{P}^*y\|_* \leq \max\{\|x\|_*, \|y\|_*\}$, where $\|\cdot\|_*$ is the conjugate norm and $P^*$ is the conjugate mapping.

Clearly, A.1) and A.2) are exactly Assumptions (A1) and (A2). Moreover, A.3) and Assumption (A4) are equivalent, since both are equivalent to the decomposability property (DP), see [214, Lemma B.1] and Lemma 2.7. Furthermore, if Assumption (A3) does not hold, ($\mathrm{NSP}^{\mathcal{C}}$) is only a sufficient condition. For $v \in \mathcal{C}+(-\mathcal{C})$, the minimal decomposition is uniquely given by $\hat{v}^{(1)} = v$ and $\hat{v}^{(2)} = 0$, since $\mathcal{C} = \mathcal{X}$. Thus, ($\mathrm{NSP}^{\mathcal{C}}$) is equivalent to the sufficient condition in [137, Lemma 3.1], namely

$$\|PBv\| < \|\overline{P}Bv\| \tag{2.5}$$

for all $P \in \mathcal{P}_s$ and all $v \in \mathrm{null}(A)$, $Bv \neq 0$. If additionally Assumption (A3) holds, then (2.5) and ($\mathrm{NSP}^{\mathcal{C}}$) are also necessary conditions.

Specific NSPs for all the settings in our running examples introduced in Example 2.3 are already known in the literature. We will now derive these known NSPs from the generalized null space property ($\mathrm{NSP}^{\mathcal{C}}$). Recall that Assumptions (A1) to (A3) are satisfied in all four settings of Example 2.3, so that only Assumption (A4) needs to be checked. It is worth mentioning that in these settings, the null space characterization leads to tractable algorithms to recover $x^{(0)}$, by using linear programming or semidefinite programming to minimize the $\ell_1$-norm or the nuclear norm $\|\cdot\|_*$, respectively. However, already in the case of recovering sparse vectors it is $\mathcal{NP}$-hard to check the classical null space property for a given sensing matrix $A$, as shown by Tillmann and Pfetsch [237].

**Example 2.12.**

(2.12.1) *Recovery of sparse vectors by $\ell_1$-minimization, Example (2.3.1) continued*
Recall from Example (2.3.1) that in this case, $\mathcal{P}$ is the set of orthogonal projectors onto all coordinate subspaces $\mathcal{E}_S$, with $S \subseteq [n]$ and $PBx = x_S$. The decomposability property (DP) is satisfied, since

$$\|v_S\|_1 + \|v_{\overline{S}}\|_1 = \|v\|_1$$

holds for all $v \in \mathbb{R}^n$ and all $S \subseteq [n]$. Since $\mathcal{C} = \mathcal{X}$, (DP) is equivalent to Assumption (A4) by Lemma 2.7. Thus, ($\mathrm{NSP}^{\mathcal{C}}$) characterizes uniform recovery by Theorem 2.10. For $v \in \mathbb{R}^n$, the decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathbb{R}^n$ and $\|\hat{v}^{(2)}\|_1$ minimal is unique and given by $\hat{v}^{(1)} = v$ as well as $\hat{v}^{(2)} = 0$. Consequently, Condition ($\mathrm{NSP}^{\mathcal{C}}$) simplifies to the regular null space property (NSP) (see, e.g., Foucart and Rauhut [104]):

$$\|v_S\|_1 < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \mathrm{null}(A) \setminus \{0\},\ \forall\, S \subseteq [n],\ |S| \leq s. \tag{NSP}$$

Note that $-v_{\overline{S}} \in \mathcal{D}$ holds trivially for all $v \in \mathbb{R}^n$, since $\mathcal{D} = \mathbb{R}^n$.

(2.12.2) *Recovery of sparse nonnegative vectors by $\ell_1$-minimization, Example (2.3.2) continued*
For the recovery of sparse nonnegative vectors, i.e., if the additional side con-

straint $x \geq 0$ is present, Assumption (A4) is satisfied, since for all $x \in \mathbb{R}^n_+$, we have $\|x\|_1 = \mathbb{1}^\top x$. Since the decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathbb{R}^n_+$ and minimal $\|\hat{v}^{(2)}\|_1$ is unique and given by $\hat{v}^{(1)} = v^+$ and $\hat{v}^{(2)} = v^-$, the general null space property (NSP$^{\mathcal{C}}$) of order $s$ for the set $\mathcal{C}$ is equivalent to the known nonnegative null space property [143, 256]:

$$v_{\overline{S}} \leq 0 \implies \sum_{i \in S} v_i < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \operatorname{null}(A)\backslash\{0\},\ \forall\, S \subseteq [n],\ |S| \leq s, \tag{NSP$_{\geq 0}$}$$

since $\mathcal{P}$ is again the set of orthogonal projectors onto all coordinate subspaces $\mathcal{E}_S$, with $S \subseteq [n]$ and $PBx = x_S$.

(2.12.3) *Recovery of low-rank matrices by nuclear norm minimization, Example (2.3.3) continued*
The nuclear norm also satisfies

$$\|V\|_* = \|PV\|_* + \|\overline{P}V\|_*$$

for all $V \in \mathbb{R}^{n_1 \times n_2}$ and all $P \in \mathcal{P}$ as defined in Example (2.3.3). Thus, (DP) is satisfied, and since $\mathcal{C} = \mathcal{X}$, Assumption (A4) is satisfied as well by Lemma 2.7. This implies that the general null space property (NSP$^{\mathcal{C}}$) characterizes uniform recovery for low-rank matrices. As in the case of sparse vectors, the decomposition $V = \hat{V}^{(1)} - \hat{V}^{(2)}$ with $\hat{V}^{(1)}$, $\hat{V}^{(2)} \in \mathbb{R}^{n_1 \times n_2}$ and $\|\hat{V}^{(2)}\|_*$ minimal is unique and given by $\hat{V}^{(1)} = V$ as well as $\hat{V}^{(2)} = 0$, so that Condition (NSP$^{\mathcal{C}}$) simplifies to the well-known NSP [192, 209], [104, Theorem 4.40]:

$$\sum_{j \in S} \sigma_j(V) < \sum_{j \in \overline{S}} \sigma_j(V) \quad \forall\, V \in \operatorname{null}(A)\backslash\{0\},\ \forall\, S \subseteq [n_{\min}],\ |S| \leq s, \tag{NSP$^*$}$$

where $n_{\min} = \min\{n_1, n_2\}$, $\sigma(V)$ is the vector of singular values of $V$, and $S$ is the index set associated to a projection $P \in \mathcal{P}$. For symmetric matrices $X \in \mathcal{S}^n$ this simplifies to

$$\|\lambda_S(V)\|_1 < \|\lambda_{\overline{S}}(V)\|_1 \quad \forall\, V \in \operatorname{null}(A)\backslash\{0\},\ \forall\, S \subseteq [n],\ |S| \leq s,$$

where $\lambda(V)$ is the vector of eigenvalues of $V$.

(2.12.4) *Recovery of low-rank positive semidefinite matrices by nuclear norm minimization, Example (2.3.4) continued*
Again, under the additional side constraint $X \succeq 0$, Assumption (A4) is satisfied as well. For a matrix $V \in \mathcal{S}^n$ with eigenvalue decomposition $V = U \operatorname{Diag}(\lambda)\, U^\top$, where $\lambda$ is the vector of eigenvalues and $U \in \mathcal{O}^n$, define matrices $V^+ \in \mathcal{S}^n_+$

and $V^- \in \mathcal{S}^n_+$ as $V^+ = U \operatorname{Diag}(\lambda^+)\, U^\top$ and $V^- = U \operatorname{Diag}(\lambda^-)\, U^\top$. Analogously to the case of sparse nonnegative vectors, the decomposition $V = \hat{V}^{(1)} - \hat{V}^{(2)}$ with $\hat{V}^{(1)}, \hat{V}^{(2)} \in \mathcal{S}^n_+$ and $\|\hat{V}^{(2)}\|_*$ minimal is unique and given by $\hat{V}^{(1)} = V^+$ and $\hat{V}^{(2)} = V^-$. Thus, the general null space property (NSP$^{\mathcal{C}}$) simplifies to the following NSP [146, 192]:

$$\begin{gathered}\lambda_{\overline{S}}(V) \leq 0 \implies \sum_{j \in S} \lambda_j(V) < \|\lambda_{\overline{S}}(V)\|_1 \\ \forall\, V \in (\operatorname{null}(A) \cap \mathcal{S}^n)\backslash\{0\},\ \forall\, S \subseteq [n],\ |S| \leq s,\end{gathered} \tag{NSP$^*_{\succeq 0}$}$$

where $\lambda(V)$ is the vector of eigenvalues of $V$.

**Remark 2.13.** The left hand side in the formulation of the nonnegative null space property (NSP$_{\geq 0}$) in Example (2.12.2) satisfies $\sum_{i \in S} v_i \leq \|v_S\|_1$. Additionally, if $v_{\overline{S}} > 0$ for some $v \in \operatorname{null}(A) \setminus \{0\}$, then the inequality $\sum_{i \in S} v_i < \|v_{\overline{S}}\|_1$ need not hold for this particular $v$, see Example 3.11 for an explicit case. This already indicates that (NSP$_{\geq 0}$) is weaker than (NSP).

The simple observation in Remark 2.13 already indicates that imposing side constraints leads to weaker NSPs, which implies that it is more likely for a sensing map to satisfy the resulting NSP. This underlines the importance of incorporating additional structure in form of side constraints into the recovery problem.

**Remark 2.14.** The condition in (NSP$^{\mathcal{C}}$) can be interpreted from the viewpoint of an *ordered* vector space: Let $(V, \leq)$ be a finite-dimensional ordered real vector space, i.e., a finite-dimensional real vector space $V$ with a partial order $\leq$. The *positive cone*

$$C_V \coloneqq \{x \in V \,:\, x \geq 0\}$$

is a convex cone with $C_V \cap (-C_V) = \{0\}$. If $C_V$ is full-dimensional, we have $C_V - C_V = V$ due to the next lemma. Simple examples for a full-dimensional positive cone $C_V$ are $\mathbb{R}^n$ with the usual ordering on vectors, or the space of symmetric real $n \times n$-matrices with the usual Löwner partial order: $A \preceq B :\iff B - A \succeq 0$.

**Lemma 2.15.** *Let $K \subseteq \mathbb{R}^n$ be a convex cone. Then $K - K = \mathbb{R}^n$ if and only if $K$ is full-dimensional.*

*Proof.* Clearly, if $K$ is not full-dimensional, then $K - K$ is not full-dimensional. A proof of the converse direction appears in, e.g., Ahmadi and Hall [7, Lemma 1], which we state here for completeness. Let $x \in \mathbb{R}^n$. If $x \in K$, then taking $v^{(1)} = x \in K$

and $v^{(2)} = 0 \in K$ shows $x = v^{(1)} - v^{(2)} \in K - K$. Thus, assume that $x \notin K$. Then there exists $v \in \operatorname{int}(K)$, where $\operatorname{int}(K)$ denotes the interior of $K$. This implies that there exists $\alpha \in (0, 1)$ with $w = (1 - \alpha)x + \alpha v \in K$. Rewriting this equality yields

$$x = \frac{1}{1-\alpha} w - \frac{\alpha}{1-\alpha} v \in K - K,$$

since $v$, $w \in K$ and $K$ is a cone. □

Thus, if $\mathcal{C} = C_V$ is a full-dimensional cone in $\mathbb{R}^n$, the null space property (NSP$^{\mathcal{C}}$) simplifies a bit: the condition $v \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ can be replaced by $v \in \operatorname{null}(A)$. Moreover, a decomposition $v = v^{(1)} - v^{(2)}$ with $v^{(1)}$, $v^{(2)} \in \mathcal{C}$ always exists.

**Remark 2.16.** The presented framework also captures the recovery of sparse vectors using the $\ell_0$-“norm” instead of the $\ell_1$-norm in the recovery problem, that is, solving

$$\min \{\|x\|_0 \;:\; Ax = b,\; x \in \mathbb{R}^n\}. \tag{2.6}$$

It is known that every $s$-sparse $x^{(0)}$ is the unique optimal solution of (2.6) with $b = Ax^{(0)}$, if and only if every set of $2s$ columns of $A$ is linearly independent, see, e.g., Foucart and Rauhut [104, Theorem 2.13]. In terms of $\operatorname{spark}(A)$, the *spark* of $A$, which is the minimal number of linear dependent columns of $A$, this recovery condition reads $\operatorname{spark}(A) > 2s$. In the following, we demonstrate that this setting also fits into our general framework and that the general NSP simplifies to $\operatorname{spark}(A) > 2s$ in this case. First note that $\|\cdot\|_0$ is not a norm. However, the absolute homogeneity of the norm $\|\cdot\|$ is not needed in the proof of Theorem 2.10, so that the NSP characterization in Theorem 2.10 also holds true for $\|\cdot\| = \|\cdot\|_0$. Consider the situation of Example (2.3.1) with the $\ell_0$-“norm”. Then, Assumptions (A1) to (A3) are clearly satisfied as well. For all $z \in \mathbb{R}^n$ and all $S \subseteq [n]$, the $\ell_0$-norm satisfies $\|z\|_0 = \|z_S\|_0 + \|z_{\overline{S}}\|_0$, so that Assumption (A4) is satisfied by Lemma 2.7, whose proof only exploits the (reverse) triangle inequality of $\|\cdot\|$, which is satisfied by the $\ell_0$-“norm”. By Theorem 2.10, uniform recovery of every sparse $x \in \mathbb{R}^n$ using (2.6) is characterized by (NSP$^{\mathcal{C}}$). As in the case of sparse recovery using the $\ell_1$-norm in Example (2.12.1), the decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathbb{R}^n$ and $\|\hat{v}^{(2)}\|_0$ minimal is unique and given by $\hat{v}^{(1)} = v$ and $\hat{v}^{(2)} = 0$. Thus, (NSP$^{\mathcal{C}}$) becomes

$$\|v_S\|_0 < \|v_{\overline{S}}\|_0 \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\},\; S \subseteq [n] \text{ with } |S| \leq s. \tag{2.7}$$

In order to see that (2.7) is equivalent to $\operatorname{spark}(A) > 2s$, note that

$$\begin{aligned} \operatorname{spark}(A) > 2s \;&\Leftrightarrow\; \operatorname{null}(A) \cap \{z \in \mathbb{R}^n \;:\; \|z\|_0 \leq 2s\} = \{0\} \\ &\Leftrightarrow\; \|v\|_0 > 2s \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\}. \end{aligned} \tag{2.8}$$

Now assume that $\|v\|_0 > 2s$ for all $\in \mathrm{null}(A) \setminus \{0\}$ and let $v \in \mathrm{null}(A) \setminus \{0\}$ as well as $S \subseteq [n]$ with $|S| \leq s$. Then $v_S$ is $s$-sparse, so that $\|v_S\|_0 \leq s$. If $\|v_S\|_0 \geq \|v_{\overline{S}}\|_0$, then also $\|v_{\overline{S}}\|_0 \leq s$, which implies

$$\|v\|_0 = \|v_S\|_0 + \|v_{\overline{S}}\|_0 \leq 2s,$$

which contradicts the assumption $\|v\|_0 > 2s$. Consequently, $\|v_S\|_0 < \|v_{\overline{S}}\|_0$.

For the reverse implication, assume that (2.7) holds and let $v \in \mathrm{null}(A) \setminus \{0\}$. If $|\mathrm{supp}(v)| < s$, then choosing $S = \mathrm{supp}(v)$ yields $\|v_{\overline{S}}\|_0 = 0 < \|v_S\|_0$, which contradicts (2.7). Thus, $|\mathrm{supp}(v)| \geq s$, and we can choose $S \subseteq [n]$ with $|S| = s$. This yields

$$\|v\|_0 = \|v_S\|_0 + \|v_{\overline{S}}\|_0 > 2\|v_S\|_0 = 2s,$$

which shows (2.8). Altogether, we recover the well-known condition $\mathrm{spark}(A) > 2s$ for uniform recovery of sparse vectors using (2.6).

**Remark 2.17.** In Example (2.12.1) and Remark 2.16, we have seen that sparse recovery using $\ell_1$- and $\ell_0$-minimization fits into our framework and we recover the well-known recovery conditions for these cases. We now show that using the $\ell_2$-norm instead does not fit into our framework, since Assumption (A4) is violated. In the situation of Example (2.12.1) with the $\ell_2$-norm, Assumption (A4) is equivalent to (DP) by Lemma 2.7 since $C = \mathbb{R}^n$. However, for the vector $z = (1, 1, 1)^\top$ together with $S = \{1\}$, we have

$$\sqrt{3} = \|z\|_2 < \|z_S\|_2 + \|z_{\overline{S}}\|_2 = 1 + \sqrt{2},$$

which violates (DP). This implies that we cannot use any of the theorems in this section to find recovery guarantees for $\ell_2$-minimization. This counterexample holds for all $\ell_q$-norms with $q > 1$ and $0 < q < 1$ as well. As a consequence, a null space property that characterizes successful recovery of sparse vectors in our proposed framework can only be formulated for the $\ell_1$-norm and the $\ell_0$-norm, but not for any other $\ell_q$-norm with $q \notin \{0, 1\}$.

In Chapter 3, we will consider further settings that emerge from the general framework in more detail. These include recovery for sparse integral vectors, possibly with additional upper and/or lower bounds and sparse vectors with so-called "constant modulus" constraints. Moreover, we examine recovery for block-structures matrices.

In the remaining sections of this chapter, we extend the general framework for uniform recovery presented above in Section 2.2 to stable and robust uniform recovery as well as individual recovery, that is, recovery of a fixed $x^{(0)}$, instead of all $s$-sparse $x^{(0)}$.

## 2.3 Stability and Robustness in the General Framework

Exact recovery of a sparse vector can be seen as an idealized scenario. In reality, vectors are rarely sparse, but it can be assumed that they are close to a sparse vector. In this case, we need to allow for an error when recovering $x$. If the recovery error can be controlled by the distance of $x$ to sparse vectors, then the recovery process is also called *stable* with respect to the sparsity defect in the literature. Apart from stability, another important point for realistic recovery situations is the fact that measurements are almost always corrupted by noise. Consequently, the measurements $b$ are only an approximation of the vector $Ax$, with an error $|||Ax - b||| \leq \eta$, where $\eta \geq 0$ and $||| \cdot |||$ is some appropriate norm. In this case, we cannot hope to recover the original vector $x$, but only a vector $x^*$ whose representation $Bx^*$ is close to the representation $Bx$ in the norm $\|\cdot\|$. If this distance is controlled by the measurement error $\eta$, the recovery process is called *robust* with respect to measurement errors. In order to guarantee robustness, the recovery problem needs to be adapted accordingly to incorporate the error $\eta$, i.e., the constraint $Ax = b$ needs to be replaced with $|||Ax - b||| \leq \eta$. Consequently, stable recovery as described here is a special case of robust recovery, where no measurement error is assumed.

In the classical case of (almost) sparse vectors, it is known that a slightly strengthened null space property assures that $\ell_1$-minimization is stable, and a further strengthened null space property also guarantees robustness of the recovery problem

$$\min \{\|x\|_1 \, : \, |||Ax - b||| \leq \eta\},$$

where $||| \cdot |||$ is any norm on $\mathbb{R}^n$. This result has first been established by Candès et al. [39], using the restricted isometry property instead of the null space property. An explicit proof for stability and robustness under a suitable null space property appears in Foucart and Rauhut [104]. Of course, stability and robustness of the recovery problem can also be formulated for additional side constraints as well as for the problem of recovering low-rank matrices. Conditions for stable and robust recovery of sparse nonnegative vectors are derived in Juditsky et al. [136], and Kueng and Jung [149] use the classical robust null space property to treat nonnegative vectors. Stability and robustness for low-rank matrix recovery using nuclear norm minimization has been established in Candès and Plan [43], Mohan and Fazel [180] and Recht et al. [210] using the restricted isometry property. The extension of the classical stable and robust null space property to the matrix setting appears in Oymak et al. [195] and Oymak and Hassibi [193]. A strengthened version of the robust null space property for low-rank matrix recovery is established in Kabanava et al. [139]. Conditions for stable and robust recovery of low-rank positive semidefinite matrices appear in Kong et al. [146] who directly transfer the results of [136] for the recovery of sparse nonnegative vectors to matrices.

In the following, we demonstrate that the concept of stability and robustness of the recovery problem can immediately be transferred to the general framework presented in the previous sections. Since stability is a special case of robustness, we first present the result for robust recovery in Section 2.3.1 and then derive stable recovery in Section 2.3.2. Finally, in Section 2.3.3, we show that stability and robustness in the case of (nonnegative) sparse recovery and recovery of low-rank (positive semidefinite) matrices can be derived as special cases of the obtained results.

In order to include stable and robust recovery into the general framework for uniform recovery in Section 2.2, we need the following additional assumption.

(A5) For all $s \geq 0$, $P \in \mathcal{P}_s$, all $v \in \mathcal{C}+(-\mathcal{C})$ and for all decompositions $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ and $\|B\hat{v}^{(2)}\|$ minimal, there exists $Q \in \mathcal{P}_s$ and $\overline{Q} \in \mathcal{P}$ with

$$QBv = PB\hat{v}^{(1)}, \quad QB\hat{v}^{(2)} = 0, \quad \text{and} \quad \overline{Q}Bv = \overline{P}Bv - PB\hat{v}^{(2)}.$$

Assumption (A5) essentially demands that every (sparsified) minimal decomposition can directly be obtained by sparsity-inducing projections. For instance, for a sparse nonnegative vector $v \in \mathbb{R}^n_+$, its minimal decomposition is unique and given by $v = v^+ - v^-$, see Example (2.12.2). Then, for every set $S \subseteq [n]$, we can define $T := \operatorname{supp}(v^+) \cap S$ to obtain $v_T = v_S^+$ and $v_T^- = 0$.

**Remark 2.18.** Assumption (A5) explains why we use a slightly modified Assumption (A4) and null space property (NSP$^{\mathcal{C}}$) in comparison to [128], see also Remarks 2.4 and 2.9. Namely, in the situation of Example (2.3.2), i.e., the recovery of sparse nonnegative vectors with $\ell_1$-minimization, the maps $P \in \mathcal{P}$ are given by orthogonal projections onto coordinate subspaces. Now, for the decomposition $v_S = (1, -1)^\top = (2, 0)^\top - (1, 1)^\top$ there is no projection $Q \in \mathcal{P}$ with $Qv_S = (2, 0)^\top$. However, for the minimal decomposition $v_S = (1, 0)^\top - (0, 1)^\top$, the corresponding projection $Q$ is given by the projection onto the coordinate subspace of the first coordinate of $v_S$. Thus, the minimality of the decomposition in Assumption (A5) ensures that this assumption is satisfied in all relevant special cases considered in Example 2.3, as we will see in Section 2.3.3. But using Assumption (A5) implies that Assumption (A4) and the null space property (NSP$^{\mathcal{C}}$) need to be adapted accordingly, that is, using a minimal decomposition instead of any decomposition. As discussed in Remark 2.4, this implies that the split into two different versions of Assumption (A4) and ($\mathrm{NSP}^{\mathcal{C}}$) is not necessary any longer, which simplifies the exposition in comparison to [128]. For exact recovery without noise, the adaption and resulting simplification is not necessary as shown in [128], but in order to have a unified presentation for exact, stable and robust recovery, we used the adapted Assumption (A4) and the null space property ($\mathrm{NSP}^{\mathcal{C}}$) also for exact recovery.

### 2.3.1 Robust Recovery

Let us now assume that we cannot take exact measurements, and that the original $x \in \mathcal{C}$ may not be sparse. Furthermore, let $|||\cdot|||$ be an appropriate norm on $\mathbb{R}^m$. If measurements $b$ are corrupted by some sort of noise $n$ with $|||n||| \leq \eta$, i.e., $b$ can be obtained as $b = Ax + n$, or if there is a measurement error $|||Ax - b||| \leq \eta$, the general recovery problem becomes

$$\min \{\|Bx\| \,:\, |||Ax - b||| \leq \eta,\ x \in \mathcal{C}\} \tag{2.9}$$

for some $y \in \mathbb{R}^m$ and $\eta \geq 0$. Thus, exact recovery is not possible. Instead we want to control the recovery error $\|Bx^{(0)} - Bx^*\|$, that is, the distance between the representations of the original $x^{(0)} \in \mathcal{C}$ and the recovered $x^* \in \mathcal{C}$. Since for a given $x^{(0)} \in \mathcal{C}$, a solution $x^* \in \mathcal{C}$ of the recovery problem (2.9) only needs to satisfy $|||Ax^* - Ax^{(0)}||| \leq \eta$, the resulting vector $v = x^{(0)} - x^* \notin \mathrm{null}(A)$ in general. Thus, any NSP which ensures a bound on $\|Bx^{(0)} - Bx^*\|$ needs to hold for all $v \in \mathcal{C} + (-\mathcal{C})$. This in contrast to the exact recovery in Section 2.2, where the NSPs only need to hold for elements $v \in \mathrm{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$. This implies that the error bound will depend on a term $|||Av|||$, which consequently needs to be incorporated into a robust NSP as well. An additional modification needs to be done in order to also account for elements $x \in \mathcal{C}$, which are not exactly sparse, but only “close” to sparse elements. Applying these modifications to ($\mathrm{NSP}^{\mathcal{C}}$) yields the following robust NSP.

**Definition 2.19.** *The linear sensing map $A$ satisfies the* general robust null space property *of order $s$ with constants $\rho \in (0,1)$ and $\tau > 0$ for the set $\mathcal{C}$ if and only if for all $v \in \mathcal{C} + (-\mathcal{C})$ and all $P \in \mathcal{P}_s$ it holds that*

$$-\overline{P}Bv \in \mathcal{D} \implies \forall\, \hat{v}^{(1)}, \hat{v}^{(2)} \in \mathcal{C} \text{ with } v = \hat{v}^{(1)} - \hat{v}^{(2)} \text{ and } \|B\hat{v}^{(2)}\| \text{ minimal:}$$
$$\|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| \leq \rho\|\overline{P}Bv\| + \tau|||Av|||. \tag{$\mathrm{rNSP}^{\mathcal{C}}_{\rho,\tau}$}$$

Analogously to Theorem 2.10, the robust null space property ($\mathrm{rNSP}^{\mathcal{C}}_{\rho,\tau}$) can be used to bound the error of recovery using the general robust recovery problem (2.9).

**Theorem 2.20.** *Suppose that Assumptions (A1) to (A5) are satisfied and that* $\|Bv\| = \|PBv\| + \|\overline{P}Bv\|$ *holds for all* $v \in \mathcal{C} + (-\mathcal{C})$ *and all* $P \in \mathcal{P}$*. Let* $A$ *be a linear sensing map and* $s \geq 0$*. Consider the error bound*

$$-\overline{P}Bv \in \mathcal{D} \implies \|Bx - Bz\| \leq \tfrac{1+\rho}{1-\rho}\Big(\|Bz\| - \|Bx\| + 2\|\overline{P}Bx\|\Big) + \tfrac{2\tau}{1-\rho}|||A(x-z)|||, \tag{rEB}$$

*where* $x$, $z \in \mathcal{C}$, $v := x - z$ *and* $P \in \mathcal{P}_s$*. Then the linear sensing map* $A$ *satisfies the general robust null space property* ($\mathrm{rNSP}^{\mathcal{C}}_{\rho,\tau}$) *of order* $s$ *with constants* $\rho \in (0,1)$ *and* $\tau > 0$ *for the set* $\mathcal{C}$ *if and only if the error bound* (rEB) *holds for all* $x$, $z \in \mathcal{C}$ *and all* $P \in \mathcal{P}_s$*.*

*Proof.* Let $A$ be a linear sensing map, and let $s \geq 0$. Let Assumptions (A1) to (A5) be satisfied and assume that $\|Bv\| = \|PBv\| + \|\overline{P}Bv\|$ holds for all $v \in \mathcal{C} + (-\mathcal{C})$ and all $P \in \mathcal{P}$.

Suppose that the general robust null space property ($\mathrm{rNSP}^{\mathcal{C}}_{\rho,\tau}$) holds. Let $x$, $z \in \mathcal{C}$ and $P \in \mathcal{P}_s$ with $v := x - z \in \mathcal{C} + (-\mathcal{C})$ as well as $-\overline{P}Bv \in \mathcal{D}$. Furthermore, let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ and $\|B\hat{v}^{(2)}\|$ minimal. Assumption (A4) implies

$$\|PB\hat{v}^{(2)}\| + \|\overline{P}Bv\| \leq \|Bz\| - \|Bx\| + \|PB\hat{v}^{(1)}\| + 2\|\overline{P}Bx\|. \tag{2.10}$$

By Assumption (A5), there exist $Q \in \mathcal{P}_s$, $\overline{Q} \in \mathcal{P}$ with $QBv = PB\hat{v}^{(1)}$, $QB\hat{v}^{(2)} = 0$, and $\overline{Q}Bv = \overline{P}Bv - PB\hat{v}^{(2)}$. We have $-\overline{P}Bv \in \mathcal{D}$ by assumption and $PB\hat{v}^{(2)} \in \mathcal{D}$, since $\hat{v}^{(2)} \in \mathcal{C}$. This implies $-\overline{Q}Bv = PB\hat{v}^{(2)} - \overline{P}Bv \in \mathcal{D}$, since $c_1 + c_2 \in \mathcal{C}$ for all $c_1$, $c_2 \in \mathcal{C}$ by Assumption (A1). Thus, the robust null space property ($\mathrm{rNSP}^{\mathcal{C}}_{\rho,\tau}$) for $Q \in \mathcal{P}_s$ and $v \in \mathcal{C} + (-\mathcal{C})$ implies

$$\|QB\hat{v}^{(1)}\| - \|QB\hat{v}^{(2)}\| \leq \rho\|\overline{Q}Bv\| + \tau|||Av|||.$$

By definition, $QB\hat{v}^{(1)} = QBv = PB\hat{v}^{(1)}$, $QB\hat{v}^{(2)} = 0$ and $\overline{Q}Bv = \overline{P}Bv - PB\hat{v}^{(2)}$. Thus,

$$\begin{aligned} \|PB\hat{v}^{(1)}\| &\leq \rho\big(\|\overline{P}Bv - PB\hat{v}^{(2)}\|\big) + \tau|||Av||| \\ &\leq \rho\big(\|\overline{P}Bv\| + \|PB\hat{v}^{(2)}\|\big) + \tau|||Av||| \\ &\leq \rho\big(\|Bz\| - \|Bx\| + \|PB\hat{v}^{(1)}\| + 2\|\overline{P}Bx\|\big) + \tau|||Av|||, \end{aligned}$$

where we used (2.10) for the last inequality. Thus,

$$\|PB\hat{v}^{(1)}\| \leq \frac{\rho}{1-\rho}\big(\|Bz\| - \|Bx\| + 2\|\overline{P}Bx\|\big) + \frac{\tau}{1-\rho}|||Av|||, \tag{2.11}$$

since $1-\rho \in (0,1)$. Combining (2.10) and (2.11) yields

$$\begin{aligned}\|Bx - Bz\| = \|Bv\| &= \|PBv + \overline{P}Bv\| \\ &= \|PB\hat{v}^{(1)} - PB\hat{v}^{(2)} + \overline{P}Bv\| \\ &\leq \|PB\hat{v}^{(1)}\| + \|PB\hat{v}^{(2)}\| + \|\overline{P}Bv\| \\ &\leq \|Bz\| - \|Bx\| + 2\|PB\hat{v}^{(1)}\| + 2\|\overline{P}Bx\| \\ &\leq \frac{1+\rho}{1-\rho}\Big(\|Bz\| - \|Bx\| + 2\|\overline{P}Bx\|\Big) + \frac{2\tau}{1-\rho}|||Av|||,\end{aligned}$$

where we used (2.10) for the second inequality and (2.11) for the third inequality. This shows that the error bound (rEB) is satisfied.

In order to prove the reverse implication, suppose that the error bound (rEB) holds for all $x$, $z \in \mathcal{C}$ and all $P \in \mathcal{P}_s$. Let $P \in \mathcal{P}_s$ and $v \in \mathcal{C} + (-\mathcal{C})$ with $-\overline{P}Bv \in \mathcal{D}$. Let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ be a decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ so that $\|B\hat{v}^{(2)}\|$ is minimal. Since $v \in \mathcal{C} + (-\mathcal{C})$, there exists at least one such decomposition. Define

$$w_S^{(1)} := PB\hat{v}^{(1)}, \quad w_S^{(2)} := PB\hat{v}^{(2)}, \quad w_{\overline{S}} := -\overline{P}Bv,$$

with $w_S^{(1)}$, $w_S^{(2)} \in \mathcal{D}$ since by Assumption (A1), $PBx \in \mathcal{D}$ for all $x \in \mathcal{C}$. Moreover, $w_{\overline{S}} = -\overline{P}Bv \in \mathcal{D}$ holds by assumption. Since $\mathcal{D}$ is the image of $\mathcal{C}$ under $B$, there exist $\hat{v}_S^{(1)}$, $\hat{v}_S^{(2)}$, $v_{\overline{S}} \in \mathcal{C}$ with

$$B\hat{v}_S^{(1)} = w_S^{(1)} = PB\hat{v}^{(1)}, \quad B\hat{v}_S^{(2)} = w_S^{(2)} = PB\hat{v}^{(2)}, \quad Bv_{\overline{S}} = w_{\overline{S}} = -\overline{P}Bv.$$

Due to Assumption (A3), we have

$$Bv = PBv + \overline{P}Bv = PB\hat{v}^{(1)} - PB\hat{v}^{(2)} + \overline{P}Bv = B\big(\hat{v}_S^{(1)} - \hat{v}_S^{(2)} - v_{\overline{S}}\big),$$

which implies $v = \hat{v}_S^{(1)} - \hat{v}_S^{(2)} - v_{\overline{S}}$, since $B$ is injective by Assumption (A1). Since $\hat{v}_S^{(1)}$, $\hat{v}_S^{(2)}$, $v_{\overline{S}} \in \mathcal{C}$ and $c_1 + c_2 \in \mathcal{C}$ for all $c_1$, $c_2 \in \mathcal{C}$ by Assumption (A1), we have $\hat{v}_S^{(2)} + v_{\overline{S}} \in \mathcal{C}$. Define

$$x := \hat{v}_S^{(1)} \in \mathcal{C}, \quad z := \hat{v}_S^{(2)} + v_{\overline{S}} \in \mathcal{C},$$

so that $v = x - z$. Since by assumption, $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ such that $\|B\hat{v}^{(2)}\|$ is minimal, the error bound (rEB) implies that

$$\begin{aligned}\|B\hat{v}_S^{(1)} - B\hat{v}_S^{(2)} - Bv_{\overline{S}}\| \leq \tfrac{1+\rho}{1-\rho}\Big(&\|B\hat{v}_S^{(2)} + Bv_{\overline{S}}\| - \|B\hat{v}_S^{(1)}\| + 2\|\overline{P}B\hat{v}_S^{(1)}\|\Big) \\ &+ \tfrac{2\tau}{1-\rho}|||Av|||,\end{aligned}$$

which yields

$$\begin{aligned}&(1-\rho)\|B\hat{v}_S^{(1)} - B\hat{v}_S^{(2)} - Bv_{\overline{S}}\| \\ &\leq (1+\rho)\Big(\|B\hat{v}_S^{(2)} + Bv_{\overline{S}}\| - \|B\hat{v}_S^{(1)}\| + 2\|\overline{P}B\hat{v}_S^{(1)}\|\Big) + 2\tau|\!|\!|Av|\!|\!|.\end{aligned} \tag{2.12}$$

This shows

$$\begin{aligned}&(1-\rho)\Big(\|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| + \|\overline{P}Bv\|\Big) \\ &\leq (1-\rho)\Big(\|PB\hat{v}^{(1)} - PB\hat{v}^{(2)}\| + \|\overline{P}Bv\|\Big) \\ &= (1-\rho)\|PB\hat{v}^{(1)} - PB\hat{v}^{(2)} + \overline{P}Bv\| \\ &= (1-\rho)\|B\hat{v}_S^{(1)} - B\hat{v}_S^{(2)} - Bv_{\overline{S}}\| \\ &\leq (1+\rho)\Big(\|B\hat{v}_S^{(2)} + Bv_{\overline{S}}\| - \|B\hat{v}_S^{(1)}\| + 2\|\overline{P}B\hat{v}_S^{(1)}\|\Big) + 2\tau|\!|\!|Av|\!|\!| \\ &= (1+\rho)\Big(\|PB\hat{v}^{(2)} - \overline{P}Bv\| - \|PB\hat{v}^{(1)}\|\Big) + 2\tau|\!|\!|Av|\!|\!| \\ &\leq (1+\rho)\Big(\|PB\hat{v}^{(2)}\| + \|\overline{P}Bv\| - \|PB\hat{v}^{(1)}\|\Big) + 2\tau|\!|\!|Av|\!|\!|,\end{aligned}$$

where we used the assumption $\|Bv\| = \|PBv\| + \|\overline{P}Bv\|$ for the first equality, Inequality (2.12) for the second inequality, and $\overline{P}P = 0$ (Assumption (A2)) for the third equality. This results in the inequality

$$\begin{aligned}&(1-\rho)\Big(\|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| + \|\overline{P}Bv\|\Big) \\ &\leq (1+\rho)\Big(\|PB\hat{v}^{(2)}\| + \|\overline{P}Bv\| - \|PB\hat{v}^{(1)}\|\Big) + 2\tau|\!|\!|Av|\!|\!|.\end{aligned} \tag{2.13}$$

Rewriting Inequality (2.13), we obtain

$$2\|PB\hat{v}^{(1)}\| - 2\|PB\hat{v}^{(2)}\| - 2\rho\|\overline{P}Bv\| - 2\tau|\!|\!|Av|\!|\!| \leq 0,$$

which shows the general robust null space property (rNSP$^{\mathcal{C}}_{\rho,\tau}$) of order $s$ with constants $\rho$ and $\tau$ for the set $\mathcal{C}$. □

**Remark 2.21.** For the proof that the robust error bound (rEB) implies the robust null space property (rNSP$^{\mathcal{C}}_{\rho,\tau}$), we need that

$$\|Bv\| = \|PBv\| + \|\overline{P}Bv\| \tag{2.14}$$

holds for all $v \in \mathcal{C} + (-\mathcal{C})$ and all $P \in \mathcal{P}$. However, Lemma 2.7 only shows that Assumption (A4) implies (2.14) for all $v \in \mathcal{C}$ and all $P \in \mathcal{P}$. Thus, we added the slightly stronger property as assumption to the statement in Theorem 2.20. It turns out that in all settings considered throughout this thesis, (2.14) even holds for all $v \in \mathcal{X}$ and all $P \in \mathcal{P}$. In fact, we conjecture that Assumption (A4) already implies (2.14) for all $v \in \mathcal{C} + (-\mathcal{C})$ and all $P \in \mathcal{P}$.

If the general robust null space property (2.10) is satisfied, the error bound (rEB) holds for all $x$, $z \in \mathcal{C}$. If $x = x^{(0)} \in \mathcal{C}$ and $z = \tilde{x} \in \mathcal{C}$ is an optimal solution of the recovery problem (2.9) with $b = Ax^{(0)}$, then Theorem 2.20 yields an explicit error bound in terms of the measurement error $\eta$ and the distance of $x^{(0)}$ to sparse elements. To formalize this distance, we introduce the *best s-term approximation* of $x$ and its error in the following.

**Definition 2.22.** *Let $x \in \mathcal{X}$. The* error $\sigma_s(x)$ of the best $s$-term approximation *of $x$ is defined as*

$$\sigma_s(x) := \min \{\|Bx - Bz\| \,:\, z \in \mathcal{X},\, \exists\, P \in \mathcal{P}_s \text{ with } PBz = Bz\}, \tag{2.15}$$

*and any $z \in \mathcal{X}$ attaining this minimum is called a* best $s$-term approximation *of $x$.*

Definition 2.22 generalizes the usual notion of the best $s$-term approximation for sparse vectors $\sigma_s(x)_1 = \min \{\|x - z\|_1 \,:\, z \in \mathbb{R}^n \text{ is } s\text{-sparse}\}$, see e.g., Foucart and Rauhut [104, Definition 2.2]. We could also restrict to $z \in \mathcal{C}$ in (2.15), but this would lead to possibly larger values $\sigma_s(x)$, which in turn would weaken any (upper) bound involving $\sigma_s(x)$. Thus, we use $z \in \mathcal{X}$ in (2.15).

Theorem 2.20 can now be used to formulate the desired error bound in the norm $\|\cdot\|$ for an optimal solution of the general robust recovery problem (2.9), which depends on the measurement error and the distance to sparse elements.

**Theorem 2.23.** *Suppose that Assumptions (A1) to (A5) are satisfied. Let $A$ be a linear sensing map and $s \geq 0$. Let $x^{(0)} \in \mathcal{C}$, and let $\tilde{x}$ be an optimal solution of the recovery problem* (2.9) *with $b = Ax^{(0)} + e$ and $|||e||| \leq \eta$. If $A$ satisfies* (rNSP$^{\mathcal{C}}_{\rho,\tau}$) *of order $s$ with constants $0 < \rho < 1$ and $\tau > 1$, and if there exists $P \in \mathcal{P}_s$ such that the (unique) preimage of $PBx^{(0)} \in \mathcal{D}$ is a best $s$-term approximation of $x^{(0)}$ and $-\overline{P}B(x^{(0)} - \tilde{x}) \in \mathcal{D}$, then $\tilde{x}$ approximates $x^{(0)}$ with error*

$$\|Bx^{(0)} - B\tilde{x}\| \leq 2\tfrac{1+\rho}{1-\rho}\sigma_s(x^{(0)}) + \tfrac{4\tau}{1-\rho}\eta.$$

*Note that for all $x^{(0)} \in \mathcal{C}$, the preimage of $PBx^{(0)} \in \mathcal{D}$ exists by definition of $\mathcal{D}$ and is unique since $B$ is assumed to be injective by Assumption (A1).*

*Proof.* Suppose that Assumptions (A1) to (A5) are satisfied. Let $x^{(0)} \in \mathcal{C}$ and let $\tilde{x} \in \mathcal{C}$ be a minimizer of $\min\{\|Bx\| : |||b - Ax||| \leq \eta,\ x \in \mathcal{C}\}$ with $b = Ax^{(0)} + e$ and $|||e||| \leq \eta$. Then, $\|B\tilde{x}\| \leq \|Bx^{(0)}\|$. Let $P \in \mathcal{P}_s$ so that the preimage $z \in \mathcal{C}$ of $PBx^{(0)}$ is a best $s$-term approximation of $x^{(0)}$ and that $-\overline{P}B(x^{(0)} - \tilde{x}) \in \mathcal{D}$. This implies $\sigma_s(x^{(0)}) = \|Bx^{(0)} - Bz\| = \|Bx^{(0)} - PBx^{(0)}\| = \|\overline{P}Bx^{(0)}\|$ due to Assumption (A3). Define $v \coloneqq x^{(0)} - \tilde{x}$ and let $\hat{v}^{(1)}, \hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ and $\|B\hat{v}^{(2)}\|$ minimal. Since $A$ satisfies (rNSP$^{\mathcal{C}}_{\rho,\tau}$) of order $s$ with constants $0 < \rho < 1$ and $\tau > 1$ and $-\overline{P}B(x^{(0)} - \tilde{x}) \in \mathcal{D}$, the error bound (rEB) in Theorem 2.20 yields

$$\begin{aligned}\|Bx^{(0)} - B\tilde{x}\| &\leq \tfrac{1+\rho}{1-\rho}\Big(\|B\tilde{x}\| - \|Bx^{(0)}\| + 2\|\overline{P}Bx^{(0)}\|\Big) + \tfrac{2\tau}{1-\rho}|||Ax^{(0)} - A\tilde{x}|||\\ &\leq 2\tfrac{1+\rho}{1-\rho}\sigma_s(x^{(0)}) + \tfrac{4\tau}{1-\rho}\eta,\end{aligned}$$

since $|||Ax^{(0)} - A\tilde{x}||| \leq |||(Ax^{(0)} + e) - A\tilde{x}||| + |||e||| \leq 2\eta$. □

**Remark 2.24.** If $x^{(0)}$ is $s$-sparse, i.e., there exists $P \in \mathcal{P}_s$ with $PBx^{(0)} = Bx^{(0)}$, then $\sigma_s(x^{(0)}) = 0$ and the error bound in Theorem 2.23 becomes

$$\|Bx^{(0)} - B\tilde{x}\| \leq \tfrac{4\tau}{1-\rho}\eta.$$

Moreover, if the measurement error (or the noise level, respectively) satisfies $\eta = 0$, that is, the measurements are exact, then Theorem 2.23 asserts that $x^{(0)}$ is exactly reconstructed. Thus, we obtain the statement from Theorem 2.10 about uniform exact recovery.

**Remark 2.25.** The error bound in Theorem 2.23 is in terms of the norm $\|\cdot\|$, which is used in the objective function of the general robust recovery problem (2.9). In the classical cases this would be the $\ell_1$-norm or the nuclear norm. However, it is also interesting to have an estimate for the recovery error with respect to another norm, e.g., the $\ell_2$-norm or the Frobenius norm in the classical cases. This can be modeled by using a third norm on $\mathcal{E}$ and an adaption of the robust null space property (rNSP$^{\mathcal{C}}_{\rho,\tau}$).

**Unconstrained Case** Next we consider the important case where there are no additional side constraints, that is, $\mathcal{C} = \mathcal{X}$ and $\mathcal{D} = \mathcal{E}$ holds. In this case, for any $v \in \mathcal{X}$, the unique minimal decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}, \hat{v}^{(2)} \in \mathcal{X}$ is given by $\hat{v}^{(1)} = v$ and $\hat{v}^{(2)} = 0$. Thus, we can fix the minimal decomposition $v = v - 0$ in Assumption (A4), Assumption (A5), and (rEB) without loss of generality. Assumption (A4) becomes

$$\|Bx\| \leq \|Bz\| + \|PBv\| - \|\overline{P}Bv\| + 2\|\overline{P}Bx\| \tag{2.16}$$

for all $s \geq 0$, $P \in \mathcal{P}$, for all $x$, $z \in \mathcal{X}$ and $v := x - z$. Assumption (A5) simplifies to

$$\exists\, Q \in \mathcal{P}_s,\ \overline{Q} \in \mathcal{P}\ :\ QBv = PBv,\ \overline{Q}Bv = \overline{P}Bv$$

for all $P \in \mathcal{P}_s$, $v \in \mathcal{C} + (-\mathcal{C})$. Choosing $Q = P$ and $\overline{Q} = \overline{P}$ shows that Assumption (A5) is always satisfied in the unconstrained case. The error bound (rEB) becomes

$$\|Bx - Bz\| \leq \tfrac{1+\rho}{1-\rho}\Big(\|Bz\| - \|Bx\| + 2\|\overline{P}Bx\|\Big) + \tfrac{2\tau}{1-\rho}|\!|\!|A(x-z)|\!|\!| \tag{2.17}$$

for all $x$, $z \in \mathcal{X}$, $v := x - z$ and $P \in \mathcal{P}_s$. The general robust null space property (rNSP$^{\mathcal{C}}_{\rho,\tau}$) for the set $\mathcal{X}$ simplifies to the condition

$$\|PBv\| \leq \rho\|\overline{P}Bv\| + \tau|\!|\!|Av|\!|\!| \tag{2.18}$$

for all $v \in \mathcal{X}$ and all $P \in \mathcal{P}_s$. Altogether, Theorem 2.20 yields the following equivalence between the general robust null space property (2.18) and the error bound (2.17).

**Theorem 2.26.** *Let $A$ be a linear sensing map and $s \geq 0$. Suppose that Assumptions (A1) to (A3) are satisfied and that* (2.16) *holds. In case that $\mathcal{C} = \mathcal{X}$ holds, the linear sensing map $A$ satisfies the general robust null space property* (2.18) *of order $s$ with constants $\rho \in (0,1)$ and $\tau > 0$ for the set $\mathcal{X}$ if and only if the error bound* (2.17) *holds for all $x$, $z \in \mathcal{X}$ and all $P \in \mathcal{P}_s$.*

*Proof.* Since $\mathcal{C} = \mathcal{X}$, Lemma 2.7 shows that (2.16) implies $\|Bv\| = \|PBv\| + \|\overline{P}Bv\|$ for all $v \in \mathcal{X}$ and all $P \in \mathcal{P}$. Thus, the statement is exactly Theorem 2.20 restricted to $\mathcal{C} = \mathcal{X}$ and $\mathcal{D} = \mathcal{E}$. □

## 2.3.2 Stable Recovery

In the previous section, we already showed that under a strengthened null space property, a modified version of the recovery problem which also accounts for the measurement error $\eta$, guarantees robust recovery. The recovery error between the original $x^{(0)} \in \mathcal{C}$ and the recovered $x^* \in \mathcal{C}$ in the norm $\|\cdot\|$ depends on the distance of $x^{(0)}$ to sparse elements and the measurement error $\eta$. Let us now consider a special case and assume that the measurement error $\eta = 0$, that is, the measurements are exact and not corrupted by noise. Then, we again arrive at the (exact) general recovery problem (2.3), that is,

$$\min\{\|Bx\|\ :\ Ax = Ax^{(0)},\ x \in \mathcal{C}\}.$$

However, we still do not assume that $x^{(0)} \in \mathcal{C}$ is sparse. As outlined in the beginning of Section 2.3, the recovery problem (2.3) allows for stable recovery, if the error between the original $x^{(0)}$ and the recovered $x$ is controlled by the distance of $x^{(0)}$ to sparse elements. Since (2.3) is used for stable recovery, only elements $v \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ are of interest for a characterization of recovery. This implies that a corresponding error bound only needs to hold for $x, z \in \mathcal{C}$ with $Ax = Az$. Consequently, $Av = 0$, so that the terms $\tau |||Av|||$ can be omitted in the statements within the last section. This directly leads to the following stable null space property.

**Definition 2.27.** *The linear sensing map $A$ satisfies the* general stable null space property *of order $s$ with constant $\rho \in (0,1)$ for the set $\mathcal{C}$ if and only if for all $v \in (\operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C})))$ and all $P \in \mathcal{P}_s$ it holds that*

$$-\overline{P}Bv \in \mathcal{D} \implies \forall\, \hat{v}^{(1)}, \hat{v}^{(2)} \in \mathcal{C} \text{ with } v = \hat{v}^{(1)} - \hat{v}^{(2)} \text{ and } \|B\hat{v}^{(2)}\| \text{ minimal: } \\ \|PB\hat{v}^{(1)}\| - \|PB\hat{v}^{(2)}\| \leq \rho \|\overline{P}Bv\|. \qquad (\text{sNSP}^{\mathcal{C}}_{\rho})$$

Since we use the (exact) recovery problem (2.3), an error bound which ensures stability only needs to hold for all $x, z \in \mathcal{C}$ with $Ax = Az$. Thus, the term $2\tau |||A(x - z)|||$ vanishes from the robust error bound (rEB). This directly leads to the following characterization of stable recovery as a corollary of the characterization of robust recovery in Theorem 2.20.

**Corollary 2.28.** *Suppose that Assumptions (A1) to (A5) are satisfied and that $\|Bv\| = \|PBv\| + \|\overline{P}Bv\|$ holds for all $v \in \mathcal{C} + (-\mathcal{C})$ and all $P \in \mathcal{P}$. Let $A$ be a linear sensing map and $s \geq 0$. For $x, z \in \mathcal{C}$ and $P \in \mathcal{P}_s$, consider the error bound*

$$-\overline{P}Bv \in \mathcal{D} \implies \|Bx - Bz\| \leq \tfrac{1+\rho}{1-\rho}\Big(\|Bz\| - \|Bx\| + 2\|\overline{P}Bx\|\Big), \qquad (\text{sEB})$$

*where $v := x - z$. Then the linear sensing map $A$ satisfies the general stable null space property* (sNSP$^{\mathcal{C}}_{\rho}$) *of order $s$ with constant $\rho \in (0,1)$ for the set $\mathcal{C}$ if and only if the error bound* (sEB) *holds for all $x, z \in \mathcal{C}$ with $Ax = Az$ and all $P \in \mathcal{P}_s$.*

*Proof.* The statements can be obtained from Theorem 2.20 by noting that the stable null space property (sNSP$^{\mathcal{C}}_{\rho}$) only needs to hold for all $v \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ and requiring that the stable error bound (sEB) only needs to hold for all $x, z \in \mathcal{C}$ with $Ax = Az$. □

Setting $\eta = 0$ in the error bound for robust recovery in Theorem 2.23 yields the following error bound in terms of the norm $\|\cdot\|$ for the optimal solution of the general recovery problem (2.9) when the original $x^{(0)} \in \mathcal{C}$ is not $s$-sparse. The resulting error bound depends only on the distance of $x^{(0)}$ to sparse elements by using the error of the best $s$-term approximation.

**Corollary 2.29.** *Suppose that Assumptions (A1) to (A5) are satisfied. Let $A$ be a linear sensing map and $s \geq 0$. Let $x^{(0)} \in \mathcal{C}$ and let $\tilde{x}$ be an optimal solution of*

$$\min\{\|Bx\| \,:\, Ax = Ax^{(0)},\ x \in \mathcal{C}\}.$$

*If $A$ satisfies* (sNSP$^{\mathcal{C}}_{\rho}$) *of order $s$ with constant $\rho \in (0,1)$, and if there exists $P \in \mathcal{P}_s$ such that $PBx^{(0)}$ is the representation of a best $s$-term approximation of $x^{(0)}$ and $-\overline{P}B(x^{(0)} - \tilde{x}) \in \mathcal{D}$, then $\tilde{x}$ approximates $x^{(0)}$ with error*

$$\|Bx^{(0)} - B\tilde{x}\| \leq 2\tfrac{1+\rho}{1-\rho}\sigma_s(x^{(0)}).$$

*Proof.* Directly follows from Theorem 2.23 by letting $\eta = 0$. □

**Remark 2.30.** If $x^{(0)}$ is $s$-sparse, i.e., there exists $P \in \mathcal{P}_s$ with $PBx = Bx$, and if Assumption (A2) (or, to be more precise, $\overline{P}P = 0$) holds, then $\sigma_s(x^{(0)}) = 0$ and Corollary 2.29 asserts that $x^{(0)}$ is exactly recovered. Thus, we recover the statement in Theorem 2.10 about exact uniform recovery.

**Unconstrained Case** Let us again consider the important special case without side constraints, that is $\mathcal{C} = \mathcal{X}$ and $\mathcal{D} = \mathcal{E}$. Setting $\tau = 0$ and restricting to $v \in \operatorname{null}(A)$ in the simplified robust null space property for the unconstrained case in (2.18) yields the following simplified NSP for stable recovery:

$$\|PBv\| \leq \rho\|\overline{P}Bv\| \tag{2.19}$$

for all $v \in \operatorname{null}(A) \cap \mathcal{X}$ and all $P \in \mathcal{P}_s$, where $\rho \in (0,1)$. The corresponding error bound in Corollary 2.28 becomes

$$\|Bx - Bz\| \leq \tfrac{1+\rho}{1-\rho}\Big(\|Bz\| - \|Bx\| + 2\|\overline{P}Bx\|\Big), \tag{2.20}$$

where $x$, $z \in \mathcal{X}$ and $P \in \mathcal{P}_s$. Analogously to Theorem 2.26, the linear sensing map $A$ satisfies the null space property (2.19) of order $s$ with constant $\rho \in (0,1)$ if and only if the error bound (2.20) holds for all $x$, $z \in \mathcal{X}$ with $Ax = Az$ and all $P \in \mathcal{P}_s$.

### 2.3.3 Stability and Robustness for Some Special Cases

We now show that for our running examples, the statements about stable and robust recovery from the previous sections simplify to the corresponding NSPs which are already known in the literature. In all these settings, the decomposability property $\|Bv\| = \|PBv\| + \|\overline{P}Bv\|$ clearly holds for all $v \in \mathcal{X}$ and all $P \in \mathcal{P}$.

**Recovery of sparse vectors by $\ell_1$-minimization, Example (2.12.1) continued** Recall that Assumption (A4) holds and that for all $v \in \mathbb{R}^n$, the minimal decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathbb{R}^n$ is uniquely given by $\hat{v}^{(1)} = v$ and $\hat{v}^{(2)} = 0$. As already noted before, Assumption (A5) is trivially satisfied in the unconstrained case by setting $Q = P$ and $\overline{Q} = \overline{P}$. The stable null space property (sNSP$^{\mathcal{C}}_{\rho}$) of order $s$ with constant $\rho \in (0, 1)$ simplifies to the well known stable NSP (see, e.g., Foucart and Rauhut [104, Definition 4.11]):

$$\|v_S\|_1 \leq \rho \|v_{\overline{S}}\|_1$$

for all $v \in \text{null}(A)$ and all $S \subseteq [n]$ with $|S| \leq s$. Corollaries 2.28 and 2.29 become the corresponding statements in [104, Theorem 4.14, Theorem 4.12]. Similarly, the robust null space property (rNSP$^{\mathcal{C}}_{\rho,\tau}$) of order $s$ with constants $\rho \in (0, 1)$ and $\tau > 0$ simplifies to the well known robust NSP (see, e.g., [104, Definition 4.17]):

$$\|v_S\|_1 \leq \rho \|v_{\overline{S}}\|_1 + \tau |\!|\!| Av |\!|\!|$$

for all $v \in \mathbb{R}^n$ and all $S \subseteq [n]$ with $|S| \leq s$. Theorems 2.20 and 2.23 are exactly [104, Theorem 4.20, Theorem 4.19].

**Recovery of sparse nonnegative vectors by $\ell_1$-minimization, Example (2.12.2) continued** Recall that for the recovery of sparse nonnegative vectors, i.e., $\mathcal{C} = \mathbb{R}^n_+$, Assumption (A4) is satisfied and that for $v \in \mathbb{R}^n$, the decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ and $\|\hat{v}^{(2)}\|_1$ minimal is unique and given by $\hat{v}^{(1)} = v^+$, and $\hat{v}^{(2)} = v^-$. Since the supports of $v^+$ and $v^-$ are disjoint, using the projection $Q$ onto the support $\text{supp}(v^+)$ of $v^+$ and the projection $\overline{Q}$ onto $[n] \setminus \text{supp}(v^+)$ shows that Assumption (A5) is satisfied as well. Thus, stable and robust recovery is characterized by (sNSP$^{\mathcal{C}}_{\rho}$) and (rNSP$^{\mathcal{C}}_{\rho,\tau}$), respectively, which simplify to the following properties:

$$\begin{aligned}
&v_{\overline{S}} \leq 0 \implies \sum_{i \in S} v_i \leq \rho \|v_{\overline{S}}\|_1 && \forall\, v \in \text{null}(A),\ \forall\, S \subseteq [n],\ |S| \leq s,\\
&v_{\overline{S}} \leq 0 \implies \sum_{i \in S} v_i \leq \rho \|v_{\overline{S}}\|_1 + \tau |\!|\!| Av |\!|\!| && \forall\, v \in \mathbb{R}^n,\ \forall\, S \subseteq [n],\ |S| \leq s.
\end{aligned}$$

The robust nonnegative NSP and a variant of the corresponding characterization of robust recovery in Theorem 2.20 appears in Juditsky et al. [136], who treat the slightly more general setting of sign restrictions on a part of all entries of $x \in \mathbb{R}^n$.

**Recovery of low-rank matrices by nuclear norm minimization, Example (2.12.3) continued** As in the case of recovery of sparse vectors, Assumption (A4) is satisfied. Moreover, the unique minimal decomposition $V = \hat{V}^{(1)} - \hat{V}^{(2)}$ is given by $\hat{V}^{(1)} = V$ and $\hat{V}^{(2)} = 0$. Once again, setting $Q = P$ and $\overline{Q} = \overline{P}$ satisfies Assumption (A5). The stable and robust null space properties (sNSP$^{\mathcal{C}}_{\rho}$) and (rNSP$^{\mathcal{C}}_{\rho,\tau}$) simplify to

$$\sum_{j \in S} \sigma_j(V) \leq \rho \sum_{j \in \overline{S}} \sigma_j(V) \quad \forall\, V \in \operatorname{null}(A),\, \forall\, S \subseteq [n_{\min}],\, |S| \leq s,$$

$$\sum_{j \in S} \sigma_j(V) \leq \rho \sum_{j \in \overline{S}} \sigma_j(V) + \tau |||AV||| \quad \forall\, V \in \mathbb{R}^{n_1 \times n_2},\, \forall\, S \subseteq [n_{\min}],\, |S| \leq s,$$

where $n_{\min} = \min\{n_1, n_2\}$ and $\sigma(V)$ is the vector of singular values of $V$. These NSPs and the corresponding statements for stable and robust recovery can be found in [104, Exercises 4.19 and 4.20].

**Recovery of low-rank positive semidefinite matrices by nuclear norm minimization, Example (2.12.4) continued** As in the case of recovery of sparse nonnegative vectors, Assumptions (A4) and (A5) are satisfied, since for $V \in \mathcal{S}^n$ with eigenvalue decomposition $V = U \operatorname{Diag}(\lambda) W^\top$ the unique decomposition $V = \hat{V}^{(1)} - \hat{V}^{(2)}$ with $\hat{V}^{(1)}, \hat{V}^{(2)} \succeq 0$ and $\|\hat{V}^{(2)}\|_*$ minimal is given by $\hat{V}^{(1)} = U \operatorname{Diag}(\lambda^+) W^\top$ and $\hat{v}^{(2)} = U \operatorname{Diag}(\lambda^-) W^\top$, where $\lambda$ is the vector of eigenvalues of $V$. Consequently, stable and robust recovery is characterized by (sNSP$^{\mathcal{C}}_{\rho}$) and (rNSP$^{\mathcal{C}}_{\rho,\tau}$), respectively, which simplify to the following properties:

$$\lambda_{\overline{S}}(V) \leq 0 \implies \sum_{j \in S} \lambda_j(V) \leq \rho \|\lambda_{\overline{S}}(V)\|_1 \quad \forall\, V \in (\operatorname{null}(A) \cap \mathcal{S}^n),\, \forall\, S \subseteq [n],\, |S| \leq s,$$

$$\lambda_{\overline{S}}(V) \leq 0 \implies \sum_{j \in S} \lambda_j(V) \leq \rho \|\lambda_{\overline{S}}(V)\|_1 + \tau |||AV||| \quad \forall\, V \in \mathcal{S}^n,\, \forall\, S \subseteq [n],\, |S| \leq s.$$

The latter robust null space property and a variant of the corresponding characterization of robust recovery in Theorem 2.20 appears in Kong et al. [146].

**Recovery of sparse vectors by $\ell_0$-minimization** We first remark that the results within this paragraph are based on joint work with Marc E. Pfetsch. In Remark 2.16, we have seen that recovery of sparse recovery using $\ell_0$-minimization also fits into our framework, and that we recover the well-known condition $\operatorname{spark}(A) > 2s$ needed for exact uniform recovery. Using the results of Section 2.3, this condition can in principle be extended to stable and robust recovery using $\ell_0$-minimization. The argument that Assumption (A5) is trivially satisfied in the unconstrained case by setting $Q = P$ and $\overline{Q} = \overline{P}$ is also applicable to the $\ell_0$-norm. Recall from Remark 2.16

that Assumption (A4) holds. Furthermore, we choose $|||\cdot||| = \|\cdot\|_0$. Even if $\|\cdot\|_0$ is not a norm, all statements in Section 2.3 still hold for this choice, since the absolute homogeneity is not needed in the proofs. The general robust null space property ($\mathrm{rNSP}^{\mathcal{C}}_{\rho,\tau}$) with $\rho \in (0,1)$ and $\tau > 0$ simplifies to

$$\|v_S\|_0 \leq \rho \|v_{\overline{S}}\|_0 + \tau \|Av\|_0 \tag{2.21}$$

for all $v \in \mathbb{R}^n$, and all $S \subseteq [n]$ with $|S| \leq s$. By Theorem 2.20, this condition is satisfied if and only if

$$\|z - x\|_0 \leq \tfrac{1-\rho}{1+\rho}\Big(\|z\|_0 - \|x\|_0 + 2\|x_{\overline{S}}\|_0\Big) + \tfrac{2\tau}{1-\rho}\|Av\|_0$$

holds for all $x,\ z \in \mathbb{R}^n$. If $A$ satisfies the null space property (2.21) of order $s$ with constants $\rho \in (0,1)$ and $\tau > 0$, then Theorem 2.23 yields the error bound

$$\|x - \tilde{x}\|_0 \leq 2\tfrac{1+\rho}{1-\rho} \max\{0, \|x\|_0 - s\} + \tfrac{4\tau}{1-\rho}\eta \tag{2.22}$$

for a solution $\tilde{x}$ of $\min\{\|z\|_0 \,:\, Az = Ax\}$, since the best $s$-term approximation of $x$ in $\|\cdot\|_0$ is given by any $z$ with at most $s$ nonzero entries. Thus, the error $\sigma_s(x)$ of the best $s$-term approximation of $x$ is either 0, if $x$ is $s$-sparse, or $\|x\|_0 - s$, if $x$ has more than $s$ nonzero entries. Clearly, if $x$ is $s$-sparse and there is no recovery or measurement error, i.e., $\eta = 0$, then this error bound implies exact recovery. The NSP condition (2.21) is implied by a statement in terms of the spark of the matrix $A$, similar to Remark 2.16.

**Lemma 2.31.** *Let $A \in \mathbb{R}^{m\times n}$, $s \geq 0$, and $\rho \in (0,1)$ and $\tau > 0$. If* $\operatorname{spark}(A) \geq \frac{1+\rho}{\rho}s$ *and $\tau \geq s$, then $A$ satisfies the robust NSP* (2.21) *of order $s$ with constants $\rho$ and $\tau$.*

*Proof.* The proof is analogous to the exact case in Remark 2.16. Let $A \in \mathbb{R}^{m\times n}$, $s \geq 0$, and $\rho \in (0,1)$. First note that

$$\operatorname{spark}(A) \geq \tfrac{1+\rho}{\rho}s \quad \Leftrightarrow \quad \|v\|_0 \geq \tfrac{1+\rho}{\rho}s \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\}. \tag{2.23}$$

Now suppose that $\tau \geq s$ and that (2.23) holds. Let $v \in \mathbb{R}^n$ and $S \subseteq [n]$ with $|S| \leq s$. W.l.o.g. we can assume $v \neq 0$, otherwise (2.21) holds trivially. By construction, $v_S$ is $s$-sparse. If $v \notin \operatorname{null}(A)$, we have $\|Av\|_0 \geq 1$ and consequently

$$\|v_S\|_0 \leq s \leq \rho\|v_{\overline{S}}\|_0 + s\|Av\|_0 \leq \rho\|v_{\overline{S}}\|_0 + \tau\|Av\|_0,$$

i.e., (2.21) holds. Thus, we assume that $v \in \operatorname{null}(A)$. If (2.21) is violated, we have

$$\|v_S\|_0 > \rho\|v_{\overline{S}}\|_0 + \tau\|Av\|_0 = \rho\|v_{\overline{S}}\|_0.$$

This yields

$$\|v\|_0 = \|v_S\|_0 + \|v_{\overline{S}}\|_0 < s + \tfrac{s}{\rho} = \tfrac{1+\rho}{\rho} s,$$

which is a contradiction to $\|v\|_0 \geq \frac{1+\rho}{\rho} s$. This shows that (2.21) is satisfied for all $v \in \mathbb{R}^n$ and all $S \subseteq [n]$ with $|S| \leq s$. □

Even if we obtain a characterization of robust recovery when using $|||\cdot||| = \|\cdot\|_0$, this choice implies that the error bound (2.22) is weak, especially if $m$ is large, since $\|A(x - \tilde{x})\|_0 \leq m$ in general. However, if $|||\cdot|||$ is any (absolute homogeneous) norm, then the robust null space property (rNSP$^{\mathcal{C}}_{\rho,\tau}$) can never be satisfied, since taking $v \in \mathbb{R}^n$ with $v_{\overline{S}} = 0$ yields the condition

$$\|v_S\|_0 \leq \tau |||Av|||.$$

By scaling, we have $\tau|||Av||| \to 0$, but $\|v_S\|_0$ stays constant, since the norm $|||\cdot|||$ is homogeneous, whereas $\|\cdot\|_0$ is not. This implies the following important observation for the satisfiability of the robust null space properties: The condition (rNSP$^{\mathcal{C}}_{\rho,\tau}$) can only be satisfied if $\|\cdot\|$ and $|||\cdot|||$ are both compatible with respect to homogeneity. This means, the norms need to satisfy

$$\|x\| \leq \|y\| \implies \|\lambda x\| \leq \|\lambda y\| \quad \text{and} \quad |||x||| \leq |||y||| \implies |||\lambda x||| \leq |||\lambda y|||$$

for all $x$, $y$ and all $\lambda \in \mathbb{R}$. It remains an open question whether there exists an error bound for the robust NSP (2.21) where the measurement error $\eta$ is bounded in terms of a (absolute homogeneous) norm $|||\cdot|||$, such as the $\ell_2$-norm. For a related discussion, see Foucart and Rauhut [104, Remark 4.34].

In the absence of noise or recovery errors, the term $\tau\|Av\|_0$ can be omitted, and the statements above simplify as follows. We obtain the stable null space property

$$\|v_S\|_0 \leq \rho \|v_{\overline{S}}\|_0 \tag{2.24}$$

for all $v \in \operatorname{null}(A)$, where $0 < \rho < 1$, which is satisfied if and only if

$$\|z - x\|_0 \leq \tfrac{1-\rho}{1+\rho}\Big(\|z\|_0 - \|x\|_0 + 2\|x_{\overline{S}}\|_0\Big)$$

holds for all $x,\ z \in \mathbb{R}^n$ with $Ax = Az$ (c.f. Corollary 2.28). The corresponding error bound becomes

$$\|x - \tilde{x}\|_0 \leq 2\tfrac{1+\rho}{1-\rho} \max\{0, \|x\|_0 - s\},$$

where $\tilde{x}$ is a solution of $\min\{\|z\|_0 : Az = Ax\}$ (c.f. Corollary 2.29). Lastly, for stable recovery, a characterization in terms of the spark of $A$ is possible.

**Corollary 2.32.** *Let $A$ be a measurement matrix and $s \geq 0$. Then, the stable NSP* (2.24) *is equivalent to*

$$\operatorname{spark}(A) \geq \tfrac{1+\rho}{\rho} s. \tag{2.25}$$

*Proof.* The proof that (2.25) implies (2.24) is completely analogous to Lemma 2.31 above by noting that the assumption $\tau \geq s$ in Lemma 2.31 is only needed for the case $v \notin \operatorname{null}(A)$, which cannot occur for the stable NSP (2.24).

For the reverse direction, assume that (2.24) holds for all $v \in \operatorname{null}(A) \setminus \{0\}$ and all $S \subseteq [n]$ with $|S| \leq s$. Let $v \in \operatorname{null}(A) \setminus \{0\}$. If $\|v\|_0 < s$, then choosing $S = \operatorname{supp}(v)$ implies $\|v_S\|_0 = s > 0 = \rho\|v_{\overline{S}}\|_0$, which contradicts (2.24). Thus, $\|v\|_0 > s$, and we can choose $S \subseteq \operatorname{supp}(v)$ with $|S| = s$. This yields

$$\|v\|_0 = \|v_S\|_0 + \|v_{\overline{S}}\|_0 \geq \|v_S\|_0 + \tfrac{1}{\rho}\|v_S\|_0 = \tfrac{1+\rho}{\rho} s,$$

which implies (2.25) by (2.23). □

Note that the condition (2.25) is necessary and sufficient for stable recovery, whereas for robust recovery, the corresponding spark condition is only necessary. It remains an open question to also find a characterization in terms of the spark in this case.

## 2.4 Individual Recovery

Until now, we have derived conditions which guarantee that *every* sufficiently sparse element $x^{(0)} \in \mathcal{C}$ is recovered using the general recovery problem (2.3). Thus, a single measurement matrix satisfying the corresponding NSPs can be used for recovering various different elements $x^{(0)} \in \mathcal{C}$, which are only required to be sparse, but need not satisfy any other assumptions. However, in some scenarios, it is only desired to recover a single sufficiently sparse *fixed* element $x^{(0)} \in \mathcal{C}$, rather than all sufficiently sparse vectors. We call this scenario *individual recovery* in order to distinguish it from uniform recovery. Of course, any NSP is sufficient for individual recovery of a fixed sparse $x^{(0)} \in \mathcal{C}$, but these conditions are clearly too strong, as the following simple example shows.

**Example 2.33.** Consider the matrix $A \in \mathbb{R}^{3\times 3}$ defined as

$$A = \begin{pmatrix} 0 & 1 & -1 \\ 0 & -2 & 2 \\ 1 & 2 & 3 \end{pmatrix}.$$

The null space of $A$ is $\operatorname{null}(A) = \{(-5\lambda, \lambda, \lambda)^\top : \lambda \in \mathbb{R}\}$. Let $s = 1$ and consider the vector $x^{(0)} = (0, 1, 0)^\top$. The minimization problem for recovering $x^{(0)}$ reads

$$\min\{\|x\|_1 : Ax = (1, -2, 2)^\top\} = \min\left\{\|x\|_1 : x_2 = 1 - \tfrac{1}{5}x_1,\ x_3 = \tfrac{1}{5}x_1\right\}.$$

This problem has a unique global optimum at $x_1 = 0$ which yields the unique global optimal solution $\tilde{x} = (0, 1, 0)^\top = x^{(0)}$. Thus, $x^{(0)}$ has been successfully recovered. However, for $S = \{1\}$, the classical null space property (NSP) is violated, since choosing $\lambda = 1$ implies $\|v_S\|_1 = 5 > 2 = \|v_{\overline{S}}\|_1$.

Thus, we need to formulate weaker conditions in order to obtain a characterization of individual recovery. For the case of recovery of sparse vectors, corresponding conditions appear in Fuchs [107] or Tropp [240]. Individual recovery for sparse nonnegative vectors is treated by Stojnic [226] and by Lange et al. [154] under the additional integrality constraint. Similar results for recovery of low-rank (positive semidefinite) matrices have been obtained by Oymak and Hassibi [192] and Oymak et al. [194]. Chandrasekaran et al. [49] and Amelunxen et al. [9] analyze individual recovery under random measurements and present a different but very simple condition for individual recovery only based on optimality of the recovery problem.

In the previous sections on exact, stable and robust uniform recovery, we have seen that for all settings considered in the literature, the corresponding NSPs guaranteeing successful recovery can be derived as special cases from the general framework presented at the beginning of this chapter. In this section, we show that the same also holds for the above-cited NSPs known in the literature for individual recovery. In order to fit individual recovery into our general framework, let $x^{(0)} \in \mathcal{C}$ be an $s$-sparse element, and let $P \in \mathcal{P}_s$ any linear map with $PBx^{(0)} = Bx^{(0)}$. Contrary to uniform recovery treated in Sections 2.2 and 2.3, we do not need all the assumptions stated in Section 2.1. Namely, we only assume that $B$ is injective, all other Assumptions (A1) to (A3) do not have to hold, unless explicitly stated. Independently of Assumption (A4) and without exploiting any possible sparsity of $x^{(0)}$, we have the following characterization of individual recovery using the general recovery problem (2.3).

**Lemma 2.34.** *Let $B$ be injective and let $x^{(0)} \in \mathcal{C}$. Then, $x^{(0)}$ is the unique optimal solution of* (2.3) *if and only if*

$$-\big(Bv - Bx^{(0)}\big) \in \mathcal{D} \implies \|Bx^{(0)} - Bv\| > \|Bx^{(0)}\| \tag{2.26}$$

*holds for all $v \in \operatorname{null}(A)$ with $Bv \neq 0$.*

*Proof.* First, let $x^{(0)} \in \mathcal{C}$ be the unique optimal solution of (2.3). Let $v \in \operatorname{null}(A)$ with $Bv \neq 0$ and $-(Bv - Bx^{(0)}) \in \mathcal{D}$. Since $B$ is injective and $\mathcal{D}$ is the image of $\mathcal{C}$ under $B$, we have $x^{(0)} - v \in \mathcal{C}$. Since $Ax^{(0)} = A(x^{(0)} - v)$, individual recovery implies $\|Bx^{(0)} - Bv\| > \|Bx^{(0)}\|$.

For the reverse direction, let $z \in \mathcal{C}$ with $Bz \neq Bx^{(0)}$ and $Az = Ax^{(0)}$. Then, $v = x^{(0)} - z \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ and $Bv \neq 0$ as well as $-(Bv - Bx^{(0)}) = Bz \in \mathcal{D}$, since $z \in \mathcal{C}$. The NSP (2.26) implies $\|Bz\| = \|Bx^{(0)} - Bv\| > \|Bx^{(0)}\|$, which shows individual recovery of $x^{(0)}$. □

Note that $-(Bv - Bx^{(0)}) \in \mathcal{D}$ implies $x^{(0)} - v \in \mathcal{C}$, since $B$ is injective and $\mathcal{D}$ is the image of $\mathcal{C}$ under $B$. Thus, $v = x^{(0)} - (x^{(0)} - v) \in \mathcal{C} + (-\mathcal{C})$. As a consequence, we do not need to explicitly require $v \in \mathcal{C} + (-\mathcal{C})$ in Lemma 2.34.

**Remark 2.35.** We do need to assume that $x^{(0)}$ is sparse in Lemma 2.34. In fact, Lemma 2.34 can be seen as an analog of the well-known statement that for a proper convex function $f$, $x$ is the unique optimal solution of $\min\{f(x) : Ax = b\}$, if and only if the descent cone of $f$ at $x$ intersects the null space of $A$ only in $\{0\}$, see, e.g., Amelunxen et al. [9, Fact 2.8], Chandrasekaran et al. [49, Proposition 2.1], or Rudelson and Vershynin [215, Chapter 4].

We now discuss general robust individual recovery. Afterwards, we apply the results on individual recovery for our running examples from Example 2.3, and show that the NSPs known in the literature for these special case also emerge from the NSP (2.26).

**Robustness** It is also possible to formulate a version of Lemma 2.34 for robust individual recovery. Therefore, as in Section 2.3.1, we consider an appropriate norm for measuring errors on $\mathbb{R}^m$. Additionally, we can use another norm, possibly different from $\|\cdot\|$ to measure errors on $\mathcal{E}$. Therefore, let $|||\cdot|||$ be a norm on $\mathbb{R}^m$, and let $\phi(\cdot)$ be a norm on $\mathcal{E}$. Note that in Section 2.3.1, we used $\phi(\cdot) = \|\cdot\|$. Recall the general robust recovery problem (2.9), i.e.,

$$\min\{\|Bx\| \,:\, |||Ax - b||| \leq \eta,\ x \in \mathcal{C}\},$$

for some $b \in \mathbb{R}^m$ and $\eta \geq 0$. The following lemma gives a bound for the recovery error when using the general robust recovery problem to approximate a given $x^{(0)} \in \mathcal{C}$ with $b = Ax^{(0)} + e$, where $|||e||| \leq \eta$. A very similar condition in the slightly different setting of atomic norms appears in [49, Proposition 2.2].

**Lemma 2.36.** *Let $x^{(0)} \in \mathcal{C}$ be given and let $\tilde{x}$ be the solution of the general robust recovery problem* (2.9) *where $b = Ax^{(0)} + e$ with $|||e||| \leq \eta$. If $|||Av||| \geq \tau\phi(Bv)$ for all $v \in \mathcal{C} + (-\mathcal{C})$ with $-(Bv - Bx^{(0)}) \in \mathcal{D}$ and $\|Bx^{(0)} - Bv\| \leq \|Bx^{(0)}\|$, then*

$$\phi\left(Bx^{(0)} - B\tilde{x}\right) \leq \frac{2\eta}{\tau}.$$

*Proof.* Let $x^{(0)} \in \mathcal{C}$ be given and let $\tilde{x}$ be the solution of the general robust recovery problem (2.9) where $b = Ax^{(0)} + e$ with $|||e||| \leq \eta$. Assume that $|||Av||| \geq \tau\phi(Bv)$ holds for all $v \in \mathcal{C} + (-\mathcal{C})$ with $-(Bv - Bx^{(0)}) \in \mathcal{D}$ and $\|Bx^{(0)} - Bv\| \leq \|Bx^{(0)}\|$. Then, since $\tilde{x}$ is an optimal solution, we have $\|B\tilde{x}\| \leq \|Bx^{(0)}\|$. If $B\tilde{x} \neq Bx^{(0)}$, then we obtain $v := x^{(0)} - \tilde{x} \in \mathcal{C} + (-\mathcal{C})$ and $-(Bv - Bx^{(0)}) = B\tilde{x} \in \mathcal{D}$. Moreover, we have $\|Bx^{(0)} - Bv\| = \|B\tilde{x}\| \leq \|Bx^{(0)}\|$. This implies

$$\phi\left(Bx^{(0)} - B\tilde{x}\right) = \phi(Bv) \leq \frac{|||Av|||}{\tau} = \frac{|||b - A\tilde{x} - e|||}{\tau} \leq \frac{|||b - A\tilde{x}||| + |||e|||}{\tau} \leq \frac{2\eta}{\tau}. \square$$

## Individual Recovery for Some Special Cases

Let us apply the statements in this section to our running examples from Example 2.3.

**Recovery of sparse vectors by $\ell_1$-minimization, Example (2.12.1) continued** Since $v \in \text{null}(A)$ if and only if $-v \in \text{null}(A)$, the NSP in Lemma 2.34 simplifies to

$$\|x^{(0)} + v\|_1 > \|x^{(0)}\|_1 \quad \forall\, v \in \text{null}(A) \setminus \{0\},$$

which can be shown to be equivalent to the following well-known characterization for individual recovery, see Foucart and Rauhut [104, Theorem 4.30]:

$$\|v_{\overline{S}}\|_1 > |\langle v, \text{sgn}(x^{(0)})\rangle| \quad \forall\, v \in \text{null}(A) \setminus \{0\}, \tag{2.27}$$

where $x^{(0)}$ is the $s$-sparse vectors that is to be recovered, and $S = \text{supp}(x^{(0)})$. Moreover, $\text{sgn}(x)$ is the vector of the (componentwise) signs of $x$.

**Recovery of sparse nonnegative vectors by $\ell_1$-minimization, Example (2.12.2) continued** For the recovery of sparse nonnegative vectors, i.e., $\mathcal{C} = \mathbb{R}^n_+$, the following property characterizes individual recovery by Lemma 2.34:

$$x^{(0)} + v \geq 0 \implies \mathbb{1}^\top v > 0 \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\}. \tag{2.28}$$

Note that the vector $v$ has been replaced by $-v$ in Lemma 2.34 to obtain the NSP (2.28). This NSP appears in Lange et al. [154, Theorem 4.17] for the restriction to sparse integral nonnegative vectors, but the same proof without integrality restrictions also works for general sparse nonnegative vectors. Moreover, the NSP (2.28) can be shown to be equivalent to the NSP

$$v_{\overline{S}} \geq 0 \implies \mathbb{1}^\top v > 0 \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\},$$

where $S = \operatorname{supp}(x^{(0)})$, which appears in Stojnic [226, Corollary 3].

**Recovery of low-rank matrices by nuclear norm minimization, Example (2.12.3) continued** Lemma 2.34 yields the following NSP:

$$\|X^{(0)} + V\|_* > \|X^{(0)}\|_* \quad \forall\, V \in \operatorname{null}(A) \setminus \{0\},$$

which is equivalent to the known condition [192, 194]

$$\|(U^{(2)})^\top W V^{(2)}\|_* > -\operatorname{tr}\left((U^{(1)})^\top W V^{(1)}\right) \quad \forall\, W \in \operatorname{null}(A) \setminus \{0\},$$

where $X^{(0)} = [U^{(1)} U^{(2)}]\,\Sigma\,[V^{(1)} V^{(2)}]^\top$ is the full singular value decomposition of $X^{(0)}$ and $X^{(0)} = U^{(1)} \Sigma_r (V^{(1)})^\top$ is the reduced singular value decomposition of $X^{(0)}$ with $\operatorname{rank}(X^{(0)}) = r$.

**Recovery of low-rank positive semidefinite matrices by nuclear norm minimization, Example (2.12.4) continued** Lemma 2.34 proves that the following property characterizes individual recovery of low-rank positive semidefinite matrices:

$$X^{(0)} + V \succeq 0 \implies \sum_{i=1}^{n} \lambda_j(V) > 0, \tag{2.29}$$

where $\lambda(V)$ is the vector of eigenvalues of $V \in \mathcal{S}^n$. Consider the following sufficient condition for individual recovery in Oymak and Hassibi [192, Lemma 20]:

$$\forall\, W \in \operatorname{null}(A) \setminus \{0\} \;:\; \begin{cases} W \text{ not psd,} & \text{or} \\ \operatorname{tr}(W) > 0, & \text{or} \\ \eta_-\left(V^\top W V\right) > 0, \end{cases} \tag{2.30}$$

where $X^{(0)} = U\Sigma U^\top$ is the reduced eigenvalue decomposition of $X^{(0)}$, $V$ is a matrix so that $[UV]$ is unitary, and $\eta_-(W)$ denotes the number of negative eigenvalues of $W$. Then, the sufficient condition (2.30) implies the necessary and sufficient condition (2.29).

In the next chapter, we consider more interesting special cases, which did not appear as often in the literature as the special cases from our running examples. We will see that these cases also fit into our framework and that we can derive new null space properties which were not known in the literature before.

# CHAPTER 3

# Recovery Conditions for Special Cases

In this chapter, we apply the general framework from Chapter 2 to three interesting special cases which have not been treated in the literature as extensively as the cases of sparse (nonnegative) vectors or low-rank (positive semidefinite) matrices.

First of all, in Section 3.1, we start with the recovery of (positive semidefinite) block-diagonal matrices. This setting generalizes the block-sparsity structure of vectors to matrices, so that by deriving the corresponding recovery conditions for (positive semidefinite) block-diagonal matrices, we also derive well-known recovery conditions for block-sparse (nonnegative) vectors. We also present connections and differences between the "classical settings" without block-structure and the block-structured settings. All statements and results within Section 3.1 are taken from joint work with Janin Heuer, Thorsten Theobald and Marc E. Pfetsch [128].

The subsequent Section 3.2 considers the special case of recovery of integral vectors, possibly with additional box constraints or a nonnegativity constraint, in more detail. This setting appears in Keiper et al. [141], where mainly individual recovery is treated, and in Lange et al. [154], which contains a thorough analysis of individual and uniform recovery conditions for both $\ell_0$- and $\ell_1$-minimization. Since the existing conditions only deal with exact recovery, we use the general framework from Chapter 2 to derive new conditions for the case of stable and robust recovery.

Finally, in Section 3.3 we consider the recovery of sparse complex vectors $x \in \mathbb{C}^n$ with the side constraint that each nonzero entry $x_j$ has a constant modulus, i.e., there exists $c \in \mathbb{R}$ with $|x_j| \in \{0, c\}$ for all $j \in [n]$. This setting has applications in signal processing, e.g., many communication signals have this property. We derive an explicit recovery condition for sparse vectors with constant modulus constraints, which, to the best of our knowledge, is not yet known in the literature. Besides, we present an algorithmic approach to solve the resulting recovery problems. This algorithm exploits the special structure given by the constant modulus constraint. Parts of Section 3.3 are based on joint work with Tobias Fischer, Ganapati Hegde, Marius Pesavento, Marc E. Pfetsch and Andreas M. Tillmann [97] within the project "EXPRESS" (SPP 1798).

Throughout this chapter, we mostly only state results for exact uniform recovery. An easy modification according to Corollary 2.28 and Theorem 2.20 as well as Lemma 2.34 leads to corresponding results for stable, robust or individual recovery, respectively. For sparse integral vectors and vectors with constant modulus constraints we shortly mention the resulting conditions, but do not go into detail.

## 3.1 Block-Structured Vectors and Matrices

The general framework that we introduced and analyzed in Chapter 2 makes use of a representation map $B$, which can be used to formulate that an element $x$ is not sparse by itself, but rather in another representation $Bx$. However, in all explicit examples considered in the previous chapter, the representation map was trivially given by the identity map, see Example 2.3. This is due to the fact that we assumed that a (nonnegative) vector is sparse in its natural representation, which means that it consists of only a few nonzero entries, and that a (positive semidefinite) matrix by itself has only a few nonzero singular values, i.e., is low-rank. In order to obtain settings which fit into the general framework and do not use the identity map as representation map, we now consider a block-structure on vectors and matrices. To do so, we sort the entries of a vector $x \in \mathbb{R}^n$ or a matrix $X \in \mathbb{R}^{m \times n}$ into blocks, and consider the blocks as "one element". A block is considered to be zero, if all entries within are zero, and nonzero otherwise. Sparsity in this setting translates to *block-sparsity*, which means that only a few blocks contain nonzero elements. This setting of so-called *block-sparse* vectors has frequently been considered in e.g., Eldar et al. [90], Elhamifar and Vidal [91], Lin and Li [159], Stojnic et al. [230], and in, e.g., Stojnic [229] with an additional nonnegativity constraint. One important application of block-sparsity is the multiple measurement problem, see Chen and Huo [50], Cotter et al. [56], Lai and Liu [151], and van den Berg and Friedlander [242]. There, instead of a single measurement, multiple measurements of a signal are taken, and the measurements are assumed to exhibit a common sparsity structure. More generally, it can also be assumed that a vector lies in a union of (low-dimensional) subspaces. This also leads to a block-sparsity structure which can be exploited in recovery, see Blumensath and Davies [28] or Eldar and Mishali [89]. Further applications of block-structured signals include DNA multiarrays as treated in Parvaresh et al. [197], multi-band signals considered by Mishali and Eldar [179], recognition of faces in Wright et al. [253] and clustering of data in multiple subspaces, see Elhamifar and Vidal [92].

In order to recover block-sparse vectors, one possibility is to use the mixed $\ell_{p,q}$-norm $\|\cdot\|_{p,q}$ with $p$, $q > 0$, which takes the blocks into account and is defined as

$$\|x\|_{p,q} \coloneqq \Big(\sum_{i=1}^{k} \|x[i]\|_p^q\Big)^{\frac{1}{q}}.$$

Here, the entries of the vector $x \in \mathbb{R}^n$ are sorted in $k$ blocks $x[1], \dots, x[k]$. In the following, *inner norm* refers to the $\ell_p$-norm applied to each block, and *outer norm* denotes the $\ell_q$-norm applied to the vector of inner norms. As an adaption of the ordinary $\ell_1$-norm, an outer $\ell_1$-norm and an inner $\ell_2$-norm can be used, which leads to the problem

$$\min\{\|x\|_{2,1} \, : \, Ax = b, x \in \mathbb{R}^n \text{ block-structured}\}.$$

Note that neither the blocksizes need to be equal, nor the blocks need to be consecutive. Furthermore, the blocks do not need to be mutual exclusive with respect to their elements. Instead of the inner $\ell_2$-norm, any other norm can be used, since we only need to decide whether or not a block contains nonzero elements. As we will see later on, the NSP for uniform recovery of block-sparse vectors depends on the choice of the inner norm. Moreover, if all entries of $x$ are nonnegative, using an inner $\ell_1$-norm is essential to obtain an NSP for this case.

Since vectors can be seen as diagonal matrices, the block-structure can be extended to matrices as well. This leads to so-called *block-diagonal* matrices which consist of blocks along their diagonals, that is

$$X = \begin{pmatrix} X_{B_1} & & \\ & \ddots & \\ & & X_{B_k} \end{pmatrix},$$

where $X_{B_i}$ are (square) matrices. Again, we call the matrix $X$ (block-) sparse, if only a few blocks $X_{B_i}$ contain nonzero elements. Thus, we can use the mixed $\ell_{*,1}$-norm, which applies an outer $\ell_1$-norm to the vector of inner nuclear norms of the blocks, that is

$$\|X\|_{*,1} \coloneqq \sum_{i=1}^{k} \|X_{B_i}\|_*. \tag{3.1}$$

Analogously to block-sparse nonnegative vectors, a positive semidefiniteness constraint on the matrices can be added. Systems with a block-diagonal form (as formally defined in Definition 3.2) appear, e.g., in the recovery of unknown quantum states, which is also called quantum state tomography, see Eisert et al. [83] and the references therein. Frequently, quantum states are represented using low-rank Hermitian matrices. If the measurements of a quantum state are taken with an only partly-calibrated device, that is, not all calibration parameters are fully known, this introduces a sparsity structure on the calibration parameters, which can be modeled using sparse block-diagonal matrices. Apart from this application in CS, positive semidefinite block-diagonal systems appear in various other areas, which are not directly related to CS. Consider a standard semidefinite problem (SDP)

$$\min\{\langle A_0, X\rangle_{\mathrm{F}} \, : \, \langle A_p, X\rangle_{\mathrm{F}} = b_p,\ p \in \{1,\dots,m\},\ X \succeq 0\}, \tag{3.2}$$

with $A_0, \dots, A_m \in \mathcal{S}^n$, $b \in \mathbb{R}^m$ and $\langle U, V\rangle_{\mathrm{F}}$ defined in (1.1). Even if SDPs are (most of the time) theoretically solvable in polynomial time, scalability for SDPs is still a problem which often prevents SDP-based formulations to be used in practice, see also the recent survey by Majumdar et al. [170]. One approach to improve solving times for large SDPs, is to exploit sparsity in the matrices $A_p$. This can be done by introducing a block-diagonal form on $X$, corresponding to the positions of the nonzero entries in $A_p$. By reformulating the matrices in (3.2), we can obtain a block-diagonal form. The optimal solution stays unchanged, since no term related to sparsity is added to the objective function. For more information on sparsity in SDPs, see, e.g., Fukuda et al. [109], Nakata et al. [183], Vandenberghe and Andersen [244], as well as [170, Section 2].

Besides their usage for exploiting sparsity in SDPs, block-diagonal systems also appear in the analysis of structured infeasibility in SDPs. The structure of infeasible linear systems is well understood, see, e.g., Chinneck [52]. For the generalization to SDPs, an irreducible infeasible subsystem (IIS) of a semidefinite system can be defined, i.e., an infeasible subsystem such that every proper subsystem is feasible, analogously to the linear case. This leads to block-diagonal systems, and an IIS is then given by an inclusion-minimal set of infeasible block-diagonal subsystems. A full characterization of IISs is available only in the linear case, see Gleeson and Ryan [119]. For semidefinite systems, a full characterization is not available, and it turns out that subsystems with minimal block-support, i.e., *block-sparse* subsystems need to be computed in order to find an IIS, see Kellner et al. [142] for more details.

In this section, we introduce the concept of (positive semidefinite) systems in block-diagonal form and derive the corresponding NSPs and recovery statements from the general framework in Chapter 2. We also demonstrate that the well-known recovery results for block-sparse vectors can be directly obtained as a special case.

### 3.1.1 Block-Sparse (Positive Semidefinite) Matrices

In order to formally introduce the concept of block-sparsity for matrices, let $\mathcal{X} = \mathcal{S}^n$, and consider the linear sensing operator $A \colon \mathcal{S}^n \to \mathbb{R}^m$ given by

$$A(X) = (\langle A_1, X\rangle_{\mathrm{F}}, \dots, \langle A_m, X\rangle_{\mathrm{F}})^\top,$$

where $A_1, \dots, A_m \in \mathcal{S}^n$, $b \in \mathbb{R}^m$, and $X \in \mathcal{S}^n$. This results in the matrix equation $A(X) = b$.

**Remark 3.1.** In this section, we deviate from the notation used in the previous chapter and denote the image of a linear map $F$ as $F(X)$ in order to avoid confusion between matrix products and images of linear maps.

We can now define the following block-diagonal form for linear measurement operators and the appearing matrices.

**Definition 3.2.** *Let $k \geq 1$ and $B_1, \dots, B_k \neq \varnothing$ be a partition of the set $[n]$, that is, $\bigcup_{i=1}^n B_i = [n]$ with pairwise disjoint blocks $B_i$. A linear operator $A(X)$ is said to be in* block-diagonal form *with blocks $B_1, \dots, B_k$ if and only if $(A_i)_{s,t} = 0$ holds for all $(s,t) \notin (B_1 \times B_1) \cup \dots \cup (B_k \times B_k)$ and all $i \in [m]$.*

For a matrix $X \in \mathcal{S}^n$ and an index set $I \subseteq [n]$, the submatrix containing rows and columns of $X$ indexed by $I$ is denoted by $X_I$. Moreover, $\mathcal{S}^I$ (and $\mathcal{S}^I_+$) denotes the space of symmetric (positive semidefinite) $|I| \times |I|$ matrices with rows and columns indexed by the elements of $I$.

In order to formulate the setting of block-diagonal matrices in the general framework from Chapter 2, let $\mathcal{E} = \mathcal{S}^{B_1} \times \dots \times \mathcal{S}^{B_k}$. We write $X \in \mathcal{E}$ as

$$X = \begin{pmatrix} X_{B_1} & & \\ & \ddots & \\ & & X_{B_k} \end{pmatrix} \text{ with } X_{B_i} \in \mathcal{S}^{B_i} \text{ for all } i \in [k].$$

Therefore, the representation map $B \colon \mathcal{X} \to \mathcal{E}$ takes $X \in \mathcal{X} = \mathcal{S}^n$ and generates $(X_{B_1}, \dots, X_{B_k})^\top$ defined as $X_{B_i} \coloneqq \{(X_{rs})_{r,\, s \in B_i}\}$ for $i \in [k]$. Note that entries outside of the blocks are ignored. The projections, which induce sparsity, are defined as $\mathcal{P} = \{P_I : I \subseteq [k]\}$, where $P_I \colon \mathcal{E} \to \mathcal{E}$ is the orthogonal projection onto the subspace $\mathcal{E}_I \coloneqq \{X \in \mathcal{E} : X_{B_i} = 0 \; \forall\, i \notin I\}$. For $P_I \in \mathcal{P}$ define its nonnegative weight as $\nu(P) = |I|$ and $\overline{P} \coloneqq P_{[k]\setminus I}$. Lastly, let the norm $\|\cdot\|$ be the mixed $\ell_{*,1}$-norm, as defined in (3.1), where $\|\cdot\|_*$ is the nuclear norm on $\mathcal{S}^{B_i}$. An element $X \in \mathcal{X}$ is called

*$s$-block-sparse*, if and only if there exists an index set

$$I \subseteq [k] \text{ with } |I| \leq s \text{ and } P_I(B(X)) = B(X),$$

which implies that $X_{B_i} = 0$ for all $i \notin I$. This yields a block-sparsity setting for matrices. Additionally, an important side constraint on the matrix $X$, which shall be recovered is given by $X \succeq 0$. This can be incorporated into the general framework from Chapter 2 by letting $\mathcal{C} = \mathcal{S}^n_+$, which implies $\mathcal{D} = \mathcal{S}^{B_1}_+ \times \cdots \times \mathcal{S}^{B_k}_+$, and the general recovery problem (2.3) simplifies to the convex optimization problem

$$\min\{\|X\|_{*,1} \, : \, A(X) = b, \; X \succeq 0\}. \tag{3.3}$$

Using the $\ell_0$-norm $\|x\|_0$ of a vector $x \in \mathbb{R}^n$, the number of nonzero blocks in a block-diagonal matrix $X \in \mathcal{S}^n$ can be written as

$$\|X\|_{*,0} = \|(\|X_{B_1}\|_*, \ldots, \|X_{B_k}\|_*)^\top\|_0.$$

Thus, the following problem finds solutions of $A(X) = b$ with minimal block support:

$$\min\{\|X\|_{*,0} \, : \, A(X) = b, \; X \succeq 0\}. \tag{3.4}$$

As in the case of sparse vectors, Problem (3.3) is a convex approximation of (3.4). This directly leads to the question when it is possible to recover a block-sparse positive semidefinite matrix $X^{(0)}$ with $\|X^{(0)}\|_{*,0} \leq s$, from $b = A(X^{(0)})$ using the convex relaxation (3.3). For the answer, we formulate a null space property in the next definition. Theorem 3.4 shows that this NSP characterizes uniform recovery using (3.3) by deriving the proposed NSP from the general framework in Chapter 2.

**Definition 3.3.** *A linear operator $A(X)$ in block-diagonal form satisfies the* semidefinite block-matrix null space property *of order $s$ if and only if*

$$V_{B_i} \preceq 0 \;\forall\, i \in \overline{S} \quad \Longrightarrow \quad \sum_{i \in S} \mathbb{1}^\top \lambda(V_{B_i}) < \sum_{i \in \overline{S}} \|V_{B_i}\|_* \tag{NSP$^*_{*,1,\succeq 0}$}$$

*holds for all $V \in (\mathrm{null}(A) \cap \mathcal{S}^n) \setminus \{0\}$ and all $S \subseteq [k]$, $|S| \leq s$, where $\lambda(V_{B_i})$ is the vector of eigenvalues of $V_{B_i}$.*

**Theorem 3.4.** *Let $A(X)$ be a linear operator in block-diagonal form and $s \geq 0$. The following statements are equivalent:*

*(i) Every $X^{(0)} \in \mathcal{S}^n_+$ with $\|X^{(0)}\|_{*,0} \leq s$ is the unique optimal solution of* (3.3) *with $b = A(X^{(0)})$.*

*(ii) $A(X)$ satisfies the semidefinite block-matrix null space property of order $s$.*

*Proof.* In the situation described above, using $\mathcal{C} = \mathcal{S}_+^n$, $\mathcal{D} = \mathcal{S}_+^{B_1} \times \cdots \times \mathcal{S}_+^{B_k}$ and the mixed $\ell_{*,1}$-norm Assumptions (A1) to (A3) are clearly satisfied. In order to prove that Assumption (A4) holds, let $P \coloneqq P_S \in \mathcal{P}_s$ and $Z$, $X \in \mathcal{S}_+^n$ with $V = X - Z$. Moreover, let $\hat{V}^{(1)}$, $\hat{V}^{(2)} \in \mathcal{S}_+^n$ with $V = \hat{V}^{(1)} - \hat{V}^{(2)}$ be a minimal decomposition. By definition of $\mathcal{P}$ and $\|\cdot\|_{*,1}$, we have $\|B(X)\|_{*,1} = \|P(B(X))\|_{*,1} + \|\overline{P}(B(X))\|_{*,1}$. Furthermore,

$$\sum_{i=1}^{n} \lambda_i\big(P(B(\hat{V}^{(1)}))\big) - \sum_{i=1}^{n} \lambda_i\big(P(B(\hat{V}^{(2)}))\big) = \sum_{i=1}^{n} \lambda_i\big(P(B(V))\big).$$

Thus, $\|P(B(\hat{V}^{(1)}))\|_{*,1} - \|P(B(\hat{V}^{(2)}))\|_{*,1} = \|P(B(X))\|_{*,1} - \|P(B(Z))\|_{*,1}$, and

$$\begin{aligned}
&\|B(Z)\|_{*,1} + \|P(B(\hat{V}^{(1)}))\|_{*,1} - \|P(B(\hat{V}^{(2)}))\|_{*,1} - \|\overline{P}(B(V))\|_{*,1} + 2\|\overline{P}(B(X))\|_{*,1} \\
&= \|\overline{P}(B(Z))\|_{*,1} + \|P(B(X))\|_{*,1} - \|\overline{P}(B(V))\|_{*,1} + 2\|\overline{P}(B(X))\|_{*,1} \\
&\geq \|\overline{P}(B(Z))\|_{*,1} + \|P(B(X))\|_{*,1} - \|\overline{P}(B(X))\|_{*,1} - \|\overline{P}(B(Z))\|_{*,1} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad + 2\|\overline{P}(B(X))\|_{*,1} \\
&= \|B(X)\|_{*,1},
\end{aligned}$$

which shows that Assumption (A4) is satisfied.

It remains to show that (NSP$^{\mathcal{C}}$) is equivalent to (NSP$^{*}_{*,1,\succeq 0}$). As in the case of low-rank positive semidefinite matrices in Example (2.12.4), the unique minimal decomposition $V = \hat{V}^{(1)} - \hat{V}^{(2)}$ with $\hat{V}^{(1)}$, $\hat{V}^{(2)} \in \mathcal{S}_+^n$ is given by $\hat{V}^{(1)} = V^+$ and $\hat{V}^{(2)} = V^-$. The matrices $V^+$ and $V^-$ are defined as in Example (2.12.4). Now, let $S \subseteq [k]$, $|S| \leq s$ and $P = P_S$ be fixed. Since

$$\sum_{i=1}^{n} \lambda_i\big(P(B(V))\big) = \|\lambda_i\big(P(B(V^+))\big)\|_1 - \|\lambda_i(P(B(V^-)))\|_1,$$

Condition (NSP$^{*}_{*,1,\succeq 0}$) clearly implies (NSP$^{\mathcal{C}}$).

For the reverse implication, let again $S \subseteq [k]$, $|S| \leq s$ and $P = P_S$ be fixed and let $V \in \mathrm{null}(A) \cap \mathcal{S}^n$ with $B(V) \neq 0$ and $\overline{P}(B(V)) \preceq 0$. Then (NSP$^{\mathcal{C}}$) implies

$$\begin{aligned}
0 &> \|P(B(V^+))\|_{*,1} - \|P(B(V^-))\|_{*,1} - \|\overline{P}(B(V))\|_{*,1} \\
&= \sum_{i=1}^{n} \lambda_i\big(P(B(V^+))\big) - \sum_{i=1}^{n} \lambda_i\big(P(B(V^-))\big) - \sum_{i=1}^{n} |\lambda_i\big(\overline{P}(B(V))\big)| \\
&= \sum_{i \in S} \mathbb{1}^\top \lambda(V_{B_i}) - \sum_{i \in \overline{S}} \|V_{B_i}\|_*,
\end{aligned}$$

which establishes (NSP$^{*}_{*,1,\succeq 0}$) and finishes the proof by Theorem 2.10. □

**Block-structured Matrices Without Positive Semidefiniteness** The situation where the additional side constraint $X \succeq 0$ is not present, can be modeled with $\mathcal{C} = \mathcal{X} = \mathcal{S}^n$ and $\mathcal{D} = \mathcal{E} = \mathcal{S}^{B_1} \times \cdots \times \mathcal{S}^{B_k}$, while $A$, $B$, $\mathcal{P}$, $\overline{P}$ and the norm $\|\cdot\|$ are defined as above. Consequently, the recovery problems (3.4) and (3.3) become

$$\min\{\|X\|_{*,0} \,:\, A(X) = b,\ X \in \mathcal{S}^n\}, \quad \text{and} \tag{3.5}$$

$$\min\{\|X\|_{*,1} \,:\, A(X) = b,\ X \in \mathcal{S}^n\}, \tag{3.6}$$

respectively. Although this setting was first introduced explicitly in [128], it implicitly appeared in the literature before. Namely, it can be obtained by combining the block/group case and the matrix case in Juditsky et al. [137]. As before, the next definition presents an NSP which characterizes uniform recovery of block-diagonal matrices which are not necessarily positive semidefinite.

**Definition 3.5.** *A linear operator $A(X)$ in block-diagonal form satisfies the* block-matrix null space property *of order $s$ if and only if*

$$\sum_{i\in S} \|V_{B_i}\|_* < \sum_{i\in \overline{S}} \|V_{B_i}\|_* \tag{NSP$^*_{*,1}$}$$

*holds for all $V \in (\text{null}(A) \cap \mathcal{S}^n)\backslash\{0\}$ and all $S \subseteq [k]$, $|S| \leq s$.*

**Theorem 3.6.** *Let $A(X)$ be a linear operator in block-diagonal form and $s \geq 0$. The following statements are equivalent:*

*(i) Every $X^{(0)} \in \mathcal{S}^n$ with $\|X^{(0)}\|_{*,0} \leq s$ is the unique optimal solution of* (3.6) *with $b = A(X^{(0)})$.*

*(ii) $A(X)$ satisfies the block-matrix null space property of order $s$.*

The proof of Theorem 3.6 is completely analogous to the proof of Theorem 3.4. It can be shown that also without the additional side constraint $X \succeq 0$, Assumption (A4) is satisfied and that (NSP$^{\mathcal{C}}$) and (NSP$^*_{*,1}$) are equivalent. Note that for $V \in \mathcal{S}^n$, the unique minimal decomposition $V = \hat{V}^{(1)} - \hat{V}^{(2)}$ with $\hat{V}^{(1)}, \hat{V}^{(2)} \in \mathcal{S}^n$ is given by $\hat{V}^{(1)} = V$ as well as $\hat{V}^{(2)} = 0$, see also Example (2.12.3). Theorem 2.10 then yields the desired result. Alternatively, as already stated, this result can be obtained by combining the block and the matrix case in Juditsky et al. [137].

**Remark 3.7.** The setting described in this section used $\mathcal{X} = \mathcal{S}^n$ and a non-overlapping block-structure, since the blocks $B_1, \ldots, B_k$ were defined to form a partition of $[n]$, which allowed for an easier presentation. As a slight generalization, consider $\mathcal{X} = \mathcal{C} = \mathbb{R}^{n_1 \times n_2}$ and possibly overlapping blocks $B_i \neq \varnothing$

with $B_1 \cup \dots \cup B_k = [n_1] \times [n_2]$. Additionally, the inner nuclear norms can be replaced by arbitrary norms on $\mathbb{R}^{B_i \times B_i}$. This also fits in our general setting described in Chapter 2, such that ($\mathrm{NSP}^*_{*,1}$) characterizes uniform recovery using

$$\min \Big\{ \sum_{i=1}^{k} \|X_{B_i}\| \;:\; A(X) = b,\ X \in \mathbb{R}^{n_1 \times n_2} \Big\}.$$

## 3.1.2 Block-Sparse (Nonnegative) Vectors

Block-sparse vectors $x$ can be seen as a special case of block-diagonal matrices. Indeed, they can be interpreted as a block-diagonal matrix $X$ which consists of blocks that are diagonal matrices as well. Then, the entries of $x$ coincide with the eigenvalues of $X$, and positive semidefiniteness of $X$ is equivalent to nonnegativity of $x$. The mixed $\ell_{*,1}$-norm becomes the mixed $\ell_{1,1}$-norm, so that Theorem 3.4 yields a characterization for the uniform recovery of block-sparse nonnegative vectors using $\ell_{1,1}$-minimization. Moreover, ($\mathrm{NSP}^*_{*,1}$) simplifies to the well-known NSP for block-sparse vectors. In this section, we shortly derive this well-known result as a special case of block-diagonal matrices. Moreover, we present an NSP for block-sparse nonnegative vectors, which has not appeared in the literature before. This NSP can be obtained from ($\mathrm{NSP}^*_{*,1,\succeq 0}$).

Block-structured vectors can be modeled in the general setup of Chapter 2 in a similar way as block-structured matrices by setting $\mathcal{X} = \mathbb{R}^n$. The block-structure is given by a partition $B_1, \dots, B_k$ of $[n]$ with nonempty sets $B_i$. For a (finite) set $I$, let $\mathbb{R}^I$ denote the space of elements with entries indexed by the elements of $I$. Then let $\mathcal{E} = \mathbb{R}^{B_1} \times \dots \times \mathbb{R}^{B_k}$ and write $y \in \mathcal{E}$ as $y = (y[1], \dots, y[k])^\top$, where $y[i] \in \mathbb{R}^{B_i}$ for all $i \in [k]$. Nonnegativity of $x$ can be modeled by $\mathcal{C} = \mathbb{R}^n_+$, which yields $\mathcal{D} = \mathbb{R}^{B_1}_+ \times \dots \times \mathbb{R}^{B_k}_+$. The representation map $B\colon \mathcal{X} \to \mathcal{E}$ maps $x \in \mathcal{C}$ to its block-structured representation $y[i] = (x_j)_{j \in B_i}$. The sparsity-inducing projections are given by $\mathcal{P} = \{P_I \,:\, I \subseteq [k]\}$, where $P_I\colon \mathcal{E} \to \mathcal{E}$ is the orthogonal projection onto the subspace $\mathcal{E}_I \coloneqq \{y \in \mathcal{E} \,:\, y[i] = 0\ \forall\, i \notin I\}$. For a projection $P_I \in \mathcal{P}$ we define its nonnegative weight as $\nu(P) = |I|$ and define $\overline{P} \coloneqq P_{[k] \setminus I}$. The norm $\|\cdot\|$ is defined as the the mixed $\ell_{1,1}$-norm $\|x\|_{1,1} = \sum_{i=1}^{k} \|y[i]\|_1$, where $y = Bx \in \mathcal{E}$ is the block-structured representation of $x \in \mathcal{X}$. Then, a vector $x \in \mathcal{X}$ is *$s$-block-sparse*, if there exists an index set $I \subseteq [k]$ with $|I| \leq s$ and $P_I Bx = Bx$, which for $y = Bx$ implies that $y[i] = 0$ for $i \notin I$. With these definitions, we arrive at the setting of recovery of block-sparse nonnegative vectors. The general recovery problem (2.3) becomes the recovery problem

$$\min \{ \|x\|_{1,1} \;:\; Ax = b,\ x \in \mathbb{R}^n_+ \}, \tag{3.7}$$

which is a convex approximation of the exact recovery problem

$$\min\{\|x\|_{1,0} \,:\, Ax = b,\ x \in \mathbb{R}^n_+\}.$$

The NSP for nonnegative block-linear systems can now be obtained as a direct corollary of Theorem 3.4 by a restriction to diagonal matrices. Note that if any inner $\ell_q$-norm other than the $\ell_1$-norm is used, Assumption (A4) is no longer satisfied. Thus, we explicitly need to use the mixed $\ell_{1,1}$-norm in order to formulate an NSP.

**Corollary 3.8.** *Consider a block-linear system $Ax = [A[1] \cdots A[k]]x = b$, where $b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m\times n}$ consists of $k$ blocks $A[i] \in \mathbb{R}^{m\times n_i}$. The following statements are equivalent:*

*(i) Every $x^{(0)} \in \mathbb{R}^n_+$ with $\|x^{(0)}\|_{1,0} \leq s$ is the unique optimal solution of* (3.7) *with $b = Ax^{(0)}$.*

*(ii) A satisfies the* nonnegative block-linear null space property of order $s$*, i.e.,*

$$v[\overline{S}] \leq 0 \quad \Longrightarrow \quad \sum_{i\in S} \mathbb{1}^\top v[i] < \sum_{i\in\overline{S}} \|v[i]\|_1 \tag{NSP$_{1,1,\geq 0}$}$$

*holds for all $v \in$* null$(A)\backslash\{0\}$ *and all $S \subseteq [k]$, $|S| \leq s$, where $v[\overline{S}] \coloneqq (v[i])_{i\in\overline{S}}$.*

**Block-sparse vectors** Without the nonnegativity constraint, let $\mathcal{C} = \mathcal{X} = \mathbb{R}^n$ and $\mathcal{D} = \mathcal{E}$, while $B$, $\mathcal{P}$, $\overline{P}$ are defined as before. This time, the norm $\|\cdot\|$ is defined as the mixed $\ell_{q,1}$-norm $\|y\|_{q,1} = \sum_{i=1}^k \|y[i]\|_q$, with $q \geq 1$ on $\mathbb{R}^{B_i}$. In the absence of the additional constraint $x \geq 0$ it is not necessary to use an inner $\ell_1$-norm for recovery, since in this case, Assumption (A4) is satisfied for any inner $\ell_q$-norm with $q \geq 1$. The exact recovery problem using a nonconvex $\ell_0$-term is

$$\min\{\|x\|_{q,0} \,:\, Ax = b,\ x \in \mathbb{R}^n\},$$

and using the $\ell_1$-norm as convex relaxation leads to the problem

$$\min\{\|x\|_{q,1} \,:\, Ax = b,\ x \in \mathbb{R}^n\}. \tag{3.8}$$

Similar to the previous section, define the *block-linear null space property of order s* as

$$\|v[S]\|_{q,1} < \|v[\overline{S}]\|_{q,1} \tag{NSP$_{q,1}$}$$

for all $v \in \text{null}(A)\backslash\{0\}$ and all $S \subseteq [k]$ with $|S| \leq s$, where again $v[S] \coloneqq (v[i])_{i\in S}$. This null space property characterizes the recovery for block-sparse vectors, which can be obtained as an immediate corollary of Theorem 3.6 by restricting to diagonal matrices. If the inner $\ell_q$-norms are given by the $\ell_2$-norm, this characterization is due to Stojnic et al. [230], who state as a remark that

> "it is reasonable to believe that the null-space characterization [...] can easily be generalized to the $\ell_p$ optimization"[2].

**Corollary 3.9.** *Let $A = [A[1] \cdots A[k]] \in \mathbb{R}^{m \times n}$ be in block-linear form with k blocks, $x = (x[1], \ldots, x[k])^\top \in \mathbb{R}^n$ and $s \geq 0$. Then, the following statements are equivalent:*

*(i) Every $x^{(0)} \in \mathbb{R}^n$ with $\|x^{(0)}\|_{q,0} \leq s$ is the unique optimal solution of* (3.8) *with $b = Ax^{(0)}$.*

*(ii) A satisfies the block-linear null space property of order s, i.e.,* ($\mathrm{NSP}_{q,1}$) *holds for all $v \in \mathrm{null}(A) \backslash \{0\}$ and all $S \subseteq [n]$ with $|S| \leq s$.*

As already stated, Corollary 3.9 directly follows as a special case from Theorem 3.6.

**Remark 3.10.** Similar to Remark 3.7, it is also possible to consider $\mathcal{X} = \mathcal{C} = \mathbb{R}^n$ and possibly overlapping blocks $B_i \neq \varnothing$ with $B_1 \cup \cdots \cup B_k = [n]$ instead of a partition $B_1, \ldots, B_k$ of $[n]$. Additionally, the inner $\ell_q$-norms could be replaced by arbitrary norms $\|\cdot\|$ on $\mathbb{R}^{B_i}$. This also fits in our general setting described in Chapter 2, such that ($\mathrm{NSP}_{q,1}$) characterizes uniform recovery using $\min \{\sum_{i=1}^{k} \|x[i]\| \, : \, Ax = b\}$.

### 3.1.3 Discussion of Block-Sparsity

In this section, we analyze and compare the null space properties derived for the recovery of block-structured (nonnegative) vectors and block-diagonal (positive semidefinite) matrices. In order to connect them to the corresponding NSPs in the respective non-block-structured settings, which served as running examples in Chapter 2, Table 3.1 subsumes all these NSPs. The third column gives the reference, if the NSP is already known in the literature, and states the corresponding theorem (resp. corollary) within this thesis. In the following, we will point out important connections between the NSPs for the eight settings considered in Table 3.1.

First of all, recall from Section 3.1.2 that the block-linear and the nonnegative block-linear cases are special cases of the block-diagonal and the semidefinite block-diagonal cases. Contrary to that, the matrix and the semidefinite matrix case are not special cases of the (semidefinite) block-diagonal cases, since the blocks are not assumed to have low rank. Nevertheless, all settings fall into the general framework in Theorem 2.10.

[2] Stojnic et al. [230, p. 3077]

**Table 3.1.** Null space properties for different settings and their references.

| Setting | NSP | Reference |
|---|---|---|
| Linear case:<br>$\min\{\|x\|_1 \,:\, Ax = b,\ x \in \mathbb{R}^n\}$ | $\|v_S\|_1 < \|v_{\overline{S}}\|_1$<br>$\forall\, v \in \mathrm{null}(A)\backslash\{0\},\ S \subseteq [n],\ \|S\| \leq s.$ | [55, 71],<br>Ex. (2.12.1) |
| Nonnegative linear case:<br>$\min\{\|x\|_1 \,:\, Ax = b,\ x \in \mathbb{R}^n_+\}$ | $v_{\overline{S}} \leq 0 \implies \sum_{i \in S} v_i < \|v_{\overline{S}}\|_1$<br>$\forall\, v \in \mathrm{null}(A)\backslash\{0\},\ S \subseteq [n],\ \|S\| \leq s.$ | [143, 256],<br>Ex. (2.12.2) |
| Block-linear case:<br>$\min\{\|x\|_{q,1} \,:\, Ax = b,\ x \in \mathbb{R}^n\}$ | $\|v[S]\|_{q,1} < \|v[\overline{S}]\|_{q,1}$<br>$\forall\, v \in \mathrm{null}(A)\backslash\{0\},\ S \subseteq [k],\ \|S\| \leq s.$ | [230],<br>Cor. 3.9 |
| Nonnegative block-linear case:<br>$\min\{\|x\|_{1,1} \,:\, Ax = b,\ x \in \mathbb{R}^n_+\}$ | $v[\overline{S}] \leq 0 \implies \sum_{i \in S} \mathbb{1}^\top v[i] < \|v[\overline{S}]\|_{1,1}$<br>$\forall\, v \in \mathrm{null}(A)\backslash\{0\},\ S \subseteq [k],\ \|S\| \leq s.$ | Cor. 3.8 |
| Matrix case:<br>$\min\{\|X\|_* \,:\, \mathcal{A}(X) = b,\ X \in \mathcal{S}^n\}$ | $\|\lambda_S(V)\|_1 < \|\lambda_{\overline{S}}(V)\|_1$<br>$\forall\, V \in \mathrm{null}(\mathcal{A})\backslash\{0\},\ S \subseteq [n],\ \|S\| \leq s.$ | [192, 209],<br>Ex. (2.12.3) |
| Semidefinite matrix case:<br>$\min\{\|X\|_* \,:\, \mathcal{A}(X) = b,\ X \in \mathcal{S}^n_+\}$ | $\lambda_{\overline{S}}(V) \leq 0 \implies \sum_{j \in S} \lambda_j(V) < \|\lambda_{\overline{S}}(V)\|_1$<br>$\forall\, V \in \mathrm{null}(\mathcal{A})\backslash\{0\},\ S \subseteq [n],\ \|S\| \leq s.$ | [146, 192],<br>Ex. (2.12.4) |
| Block-diagonal case:<br>$\min\{\|X\|_{*,1} \,:\, \mathcal{A}(X) = b,\ X \in \mathcal{S}^n\}$ | $\sum_{i \in S} \|V_{B_i}\|_* < \sum_{i \in \overline{S}} \|V_{B_i}\|_*$<br>$\forall\, V \in \mathrm{null}(\mathcal{A})\backslash\{0\},\ S \subseteq [k],\ \|S\| \leq s.$ | Thm. 3.6 |
| Semidefinite block-diagonal case:<br>$\min\{\|X\|_{*,1} \,:\, \mathcal{A}(X) = b,\ X \in \mathcal{S}^n_+\}$ | $V_{B_i} \preceq 0 \,\forall\, i \in \overline{S}$<br>$\implies \sum_{i \in S} \mathbb{1}^\top \lambda(V_{B_i}) < \sum_{i \in \overline{S}} \|V_{B_i}\|_*$<br>$\forall\, V \in \mathrm{null}(\mathcal{A})\backslash\{0\},\ S \subseteq [k],\ \|S\| \leq s$ | Thm. 3.4 |

For the null space properties in Table 3.1, we now compare the conditions that need to hold in the cases with and without the additional nonnegativity or positive semidefiniteness constraints, when the inner norms used in the respective recovery problems are identical. First note that since $\mathbb{R}^n_+ \subseteq \mathbb{R}^n$ and $\mathcal{S}^n_+ \subseteq \mathcal{S}^n$, the conditions needed for characterizing uniform recovery in the presence of nonnegativity or positive semidefiniteness are not stronger than those needed without this prior knowledge. The following example demonstrates that exploiting positive semidefiniteness indeed yields a weaker condition for uniform recovery when using the nuclear norm as inner norm in both cases.

**Example 3.11.** Let $A_1, \dots, A_4$ be the block-diagonal matrices

$$A_1 = \begin{pmatrix} 0 & & & \\ & -1 & & \\ & & -1 & 0 \\ & & 0 & 2 \end{pmatrix},\ A_2 = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & -1 & 0 \\ & & 0 & -1 \end{pmatrix},\ A_3 = \begin{pmatrix} 0 & & & \\ & -1 & & \\ & & 1 & 0 \\ & & 0 & 0 \end{pmatrix},\ A_4 = \begin{pmatrix} 0 & & & \\ & 0 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{pmatrix},$$

with blocks $B_1 = \{1\}$, $B_2 = \{2\}$ and $B_3 = \{3, 4\}$, and let $b = (-1, 0, 0, 0)^\top$. Consider

$$\min\{\|X\|_{*,0} \, : \, A(X) = b,\ X \succeq 0\},$$

where $A(X) = (\langle A_1, X\rangle_\mathrm{F}, \langle A_2, X\rangle_\mathrm{F}, \langle A_3, X\rangle_\mathrm{F}, \langle A_4, X\rangle_\mathrm{F})^\top$, see (3.4). The null space $\mathrm{null}(A) = \{V \, : \, \langle A_i, V\rangle_\mathrm{F} = 0 \text{ for } i \in [4]\}$ consists of the matrices of the form

$$V = \begin{pmatrix} 3\alpha & & & \\ & \alpha & & \\ & & \alpha & 0 \\ & & 0 & \alpha \end{pmatrix}, \text{ with } \alpha \in \mathbb{R}.$$

Since only nonzero matrices in the null space of $A$ are of interest for the NSP, $\alpha$ cannot attain the value 0. The eigenvalues of $V$ are given by $\lambda = (3\alpha, \alpha, \alpha, \alpha)^\top$. For the semidefinite block-matrix null space property ($\mathrm{NSP}^*_{*,1,\succeq 0}$) of order $s = 1$ to hold for $A$, the following implications need to be satisfied for the support sets $S \in \{\varnothing, \{1\}, \{2\}, \{3\}\}$:

$$\begin{aligned}
S = \varnothing :& \quad (3\alpha, \alpha, \alpha, \alpha)^\top \le 0 \implies 0 \ \ < 6|\alpha|,\\
S = \{1\} :& \quad (\alpha, \alpha, \alpha)^\top \ \ \le 0 \implies 3\alpha < 3|\alpha|,\\
S = \{2\} :& \quad (3\alpha, \alpha, \alpha)^\top \le 0 \implies \alpha \ \ < 5|\alpha|,\\
S = \{3\} :& \quad (3\alpha, \alpha)^\top \quad \le 0 \implies 2\alpha < 4|\alpha|.
\end{aligned}$$

These are all satisfied, since for every $V \in \mathrm{null}(A)\backslash\{0\}$, $\alpha \neq 0$ holds. However, the block-matrix null space property ($\mathrm{NSP}^*_{*,1}$) of order $s$ is violated, since for $S = \{1\}$ and $\alpha \neq 0$, we have

$$\sum_{i \in S} \|V_{B_i}\|_* = 3|\alpha| \ge 3|\alpha| = \sum_{i \in \overline{S}} \|V_{B_i}\|_*.$$

As already indicated in Remark 2.13 this example shows the important aspect that explicitly exploiting nonnegativity or positive semidefiniteness can yield stronger results for uniform recovery. In order to further strengthen this point, we explicitly construct an infinite family of examples that satisfy the nonnegative block-linear null space property ($\mathrm{NSP}_{1,1,\ge 0}$) in the next subsection. Besides highlighting the favorable effect of additional side constraints, this construction also shows that the proposed null space properties are meaningful in the sense that they are satisfied by certain general (families of) matrices.

## An infinite family satisfying the nonnegative block-linear NSP

The NSPs for the nonnegative block-linear case and for the semidefinite block-diagonal case hold in many situations. The next theorem presents a specific infinite family of instances for block sizes $(n_1, \dots, n_k) = (2, 1, \dots, 1)$, so that the nonnegative block-linear NSP holds, whereas both the (unrestricted) block-linear NSP and the nonnegative linear NSP are violated. In order to construct such a family, we employ the following characterization of the nonnegative linear NSP due to Donoho and Tanner [73].

**Proposition 3.12** (Donoho and Tanner [73]). *Let $A \in \mathbb{R}^{m \times n}$ be a matrix with nonzero columns $a^{(1)}, \dots, a^{(n)}$ and $m < n$, and let $s \geq 1$. Then $A$ satisfies the nonnegative null space property* ($\text{NSP}_{\geq 0}$) *of order $s$ if and only if the polytope $P :=$* $\operatorname{conv}\{a^{(1)}, \dots, a^{(n)}, 0\}$ *has $n+1$ vertices and is outwardly $s$-neighborly, that is, every subset of $s$ vertices not including the origin span a face of $P$.*

**Remark 3.13.** With the same preconditions, $A$ satisfies the unrestricted linear NSP of order $s$ if and only if the polytope $P' := \operatorname{conv}\{\pm a^{(1)}, \dots, \pm a^{(n)}\}$ has $2n$ vertices and is $s$-centrally neighborly, i.e., any $s$ vertices not including an antipodal pair span a face of $P$, see Donoho [67, Theorem 1] and also Foucart and Rauhut [104, Exercise 4.16]. By results of McMullen and Shephard [177], $P'$ can never be $s$-centrally neighborly for $s > \lfloor (m+1)/3 \rfloor$ (see also Donoho and Tanner [74, Section 5.3]).

**Theorem 3.14.** *Let $k > m \geq 3$ and let $B_1, \dots, B_k$ be blocks of sizes $(n_1, \dots, n_k) :=$ $(2, 1, \dots, 1)$. Define $n := \sum_{i=1}^{k} n_i = k + 1$. Then there exists a matrix $A \in \mathbb{R}^{m \times n}$ with $A = [A[1] \cdots A[k]]$ so that the nonnegative block-linear null space property* ($\text{NSP}_{1,1,\geq 0}$) *is satisfied up to the order $s^* := \lfloor m/2 - 1 \rfloor$. Moreover, for $m \geq 12$ neither the unrestricted block-linear null space property* ($\text{NSP}_{q,1}$) *of order $s^*$ is satisfied nor the nonnegative null space property* ($\text{NSP}_{\geq 0}$) *of order $s^*$ is satisfied.*

*Proof.* Let $k > m \geq 3$. Let $w^{(1)}, \dots, w^{(k-1)} \in \mathbb{R}^{m-2} \setminus \{0\}$ be $k-1$ distinct points on the moment curve $\{(t, t^2, \dots, t^{m-2})^\top : t \in \mathbb{R}\}$ in $\mathbb{R}^{m-2}$ and define the matrix $A' := [w^{(1)}, \dots, w^{(k-1)}] \in \mathbb{R}^{(m-2) \times (k-1)}$. It is well-known that the polytope $P = \operatorname{conv}\{w^{(1)}, \dots, w^{(k-1)}\}$ is a cyclic polytope, which is $\lfloor (m-2)/2 \rfloor$-neighborly, see, e.g., Ziegler [257, Corollary 0.8]. Hence, the nonnegative linear NSP of order $\lfloor (m-2)/2 \rfloor = \lfloor m/2 - 1 \rfloor$ holds for the matrix $A'$.

Let $p$ be an interior point of $P$ and set $w' = (p, 1, 0)^\top$, $w'' = (p, 0, 1)^\top$, as well as $\hat{w}^{(i)} = (w^{(i)}, 0, 0)^\top$ for $i \in [k-1]$. Let $A := [w', w'', \hat{w}^{(1)}, \dots, \hat{w}^{(k-1)}] \in \mathbb{R}^{m \times n}$ and consider the block sizes $(2, 1, \dots, 1)$. We claim that $A$ satisfies $(\mathrm{NSP}_{1,1,\geq 0})$ of order $s^*$. Namely, assume that there exists $v = (v_1, \dots, v_n)^\top \in \mathrm{null}(A) \setminus \{0\}$ and $S \subseteq [k]$ with $|S| \leq s^*$ and $v_{\overline{S}} \leq 0$ such that $\sum_{i \in S} \mathbb{1}^\top v[i] \geq \|v_{\overline{S}}\|_{1,1}$. Since $v \in \mathrm{null}(A)$ and since the penultimate and the last row of $A$ only have a single nonzero entry, we have $v_1 = v_2 = 0$. Hence, $\tilde{v} := (v_1, v_3, \dots, v_n)^\top$ is a nonzero vector in the null space of $A^\diamondsuit = [w', \hat{w}^{(1)}, \dots, \hat{w}^{(k-1)}]$ and violates the nonnegative linear NSP of order $s^*$ for $A^\diamondsuit$. However, since the polytope $P$ and thus also the polytope $\mathrm{conv}\{w', \hat{w}^{(1)}, \dots, \hat{w}^{(k-1)}\}$ are $\lfloor m/2 - 1 \rfloor$-neighborly (due to the pyramidal construction with respect to the apex $w'$), this is a contradiction.

If $m \geq 12$, the nonnegative null space property $(\mathrm{NSP}_{\geq 0})$ of order $s^*$ does not hold for $A$, because the polytope $P' := \mathrm{conv}\{w', w'', \hat{w}^{(1)}, \dots, \hat{w}^{(k-1)}\}$ is not $s^*$-neighborly. To see this, observe that any choice of vertices which includes $w'$ and $w''$ cannot span a face, hence $P'$ is not 2-neighborly, and this implies that $P'$ is not $\lfloor m/2 - 1 \rfloor$-neighborly because of $m \geq 6$.

It remains to show that $(\mathrm{NSP}_{q,1})$ of order $s^*$ is not satisfied for $m \geq 12$. Assume that it is satisfied. Then for any vector $v = (v_1, \dots, v_n)^\top \in \mathrm{null}(A) \setminus \{0\}$ and $S \subseteq [k]$ with $|S| \leq s^*$, we have $\|v[S]\|_{q,1} < \|v[\overline{S}]\|_{q,1}$. Restricting to $v_1 = 0$, the induced NSP-formula of order $s^*$ must also hold for any corresponding $(v_2, \dots, v_n)^\top \in \mathrm{null}(\tilde{A})$, where $\tilde{A}$ results from $A$ by deleting the first column, i.e., $\tilde{A} = [w'', w^{(1)} \dots, w^{(k-1)}]$. But this is a contradiction to the results of McMullen and Shephard from Remark 3.13, because we have $m \geq 12$ and thus $s^* = \lfloor m/2 - 1 \rfloor > \lfloor (m+1)/3 \rfloor$. □

**Remark 3.15.** The construction in the proof can be generalized, for example to block sizes $(n_1, \dots, n_k) = (\underbrace{2, \dots, 2}_{r}, \underbrace{1, \dots, 1}_{n-r})$ for fixed $r$ and sufficiently large $k$.

This result has two interesting implications. First, it shows that there are deterministic matrices which satisfy the NSP for block-sparse nonnegative vectors. Furthermore, its proof provides an explicit construction of such matrices. Second, the result also demonstrates that exploiting the nonnegativity as side constraint leads to successful recovery, whereas without the side constraint, successful recovery is not guaranteed. Hence, if nonnegativity is exploited, a strictly weaker NSP suffices for uniform recovery, which improves the chances of recovery.

The next section concentrates on another interesting side constraint which can be exploited in the recovery process, namely integrality.

## 3.2 Integrality Constraints on Sparse Vectors

The integrality of the vector $x^{(0)}$ which shall be recovered, is another interesting side constraint which frequently appears in applications of compressed sensing. A prominent example is discrete tomography, which is treated by Kuske et al. [150]. Moreover, integral vectors and especially binary vectors frequently appear in signal processing applications of compressed sensing, such as digital or wireless communication systems. Examples include wideband spectrum sensing in Axell et al. [11], and massive multiple-input/multiple-output (MIMO) with constellation signals, see Hegde et al. [125, 126]. The latter application deals with signals whose components are chosen from a small finite alphabet. These constellation signals appear as a result of various modulation schemes, such as $M$-phase shift keying ($M$-PSK), where an alphabet of size $M$ is used. Binary variables can be used to assign the symbols from the finite alphabet to the components of the resulting constellation signal.

An integrality condition, possibly together with additional box-constraints, can be modeled by the side constraint

$$x \in [\ell, u]_{\mathbb{Z}} \coloneqq \{x \in \mathbb{Z}^n \,:\, \ell \leq x \leq u\}, \tag{3.9}$$

where $\leq$ is applied componentwise and $\ell \in (\mathbb{R} \cup \{-\infty\})^n$ as well as $u \in (\mathbb{R} \cup \{\infty\})^n$. We assume that $\ell \leq 0 \leq u$. If the integrality constraint is directly included in the recovery program by using $\mathcal{C} = [\ell, u]_{\mathbb{Z}}$, then the resulting problem

$$\min \{\|x\|_1 \,:\, Ax = Ax^{(0)},\ x \in [\ell, u]_{\mathbb{Z}}\} \tag{3.10}$$

is nonconvex. Thus, in the literature the constraint $x \in [\ell, u]_{\mathbb{Z}}$ is commonly replaced by

$$x \in [\ell, u] \coloneqq \{x \in \mathbb{R}^n \,:\, \ell \leq x \leq u\}. \tag{3.11}$$

Recovery conditions for the important binary case with $\ell_i = 0$ and $u_i = 1$ for all $i \in [n]$ have first been derived by Stojnic [227], who presents a sufficient condition for individual recovery, that is, recovery of a fixed sparse binary vector $x^{(0)} \in \{0,1\}^n$ using $\min \{\|x\|_1 \,:\, Ax = Ax^{(0)},\ x \in [0,1]\}$. This condition is also analyzed probabilistically for (Gaussian) random matrices $A \in \mathbb{R}^{m \times n}$ to obtain thresholds for the values of measurements $m$ and the sparsity level $s$ for which, in dependence of $n$, individual recovery is possible with high probability.

For a given system of linear equations $Ax = b$, Mangasarian and Recht [172] present conditions for a vector $x^{(0)} \in \{-1,1\}^n$ with $Ax^{(0)} = b$ to be the unique optimal solution of $\min \{\|x\|_\infty \,:\, Ax = b\}$. In this context, $x^{(0)}$ is not sparse in the classical sense, but lies on a vertex of the hypercube $[-1, 1]^n$, which is the unit ball of the $\ell_\infty$-norm $\|\cdot\|_\infty$. The corresponding recovery problem replaces the $\ell_1$-norm by $\|\cdot\|_\infty$ in the objective function.

Keiper et al. [141] analyze the recovery of integral (box-constrained) vectors using a relaxed integrality constraint. The authors propose an NSP for individual recovery of a fixed $x^{(0)}$ and analyze the transition between success and failure of individual recovery for (Gaussian) random matrices $A$. These results show that exploiting the box-constraints in the recovery problem has a positive effect on the success of recovery. Results concerning individual recovery of sparse binary vectors using the convex relaxation $x \in [0, 1]^n$ for further random measurement matrices appear in Flinth and Keiper [98] and Keiper [140].

The mentioned references only treat individual recovery of a fixed vector, but not uniform recovery, and they also relax the integrality constraint for the recovery problem. Keiper et al. [141] show that if the integrality constraint in (3.9) is relaxed to (3.11), the prior knowledge of $x$ being integral does not help for recovery: uniform recovery of all sparse bounded integral $x$ is equivalent to uniform recovery of all sparse bounded $x$. This already shows that in order to exploit integrality, one has to take this into account in the recovery program. This is done in Lange et al. [154], where null space properties for uniform recovery of sparse integral (box-constrained) vectors using the $\ell_1$-minimization problem (3.10) are derived. Moreover, characterizations for uniform as well as individual recovery using the $\ell_0$-minimization problem $\min\{\|x\|_0 \,:\, Ax = Ax^{(0)},\ x \in [\ell, u]_{\mathbb{Z}}\}$ are considered.

There also exist different solution approaches for recovery of sparse binary vectors. Fosson [100] analyzes recovery conditions for a nonconvex functional, and Fosson and Abuabiah [101] propose a polynomial optimization problem for the recovery as a variant of $\ell_1$-minimization. Another modification of $\ell_1$-minimization is considered in Aïssa-El-Bey et al. [12].

In the following, we will derive the setting of recovery of sparse integral (box-constrained) vectors from the general framework presented in Chapter 2 and show that we obtain the recovery conditions presented in Lange et al. [154]. It turns out that one of the results in [154] needs to be slightly modified in order to obtain a valid characterization of uniform recovery.

The derivation of sparse integral vectors is analogous to the case of sparse vectors in Example (2.3.1). Therefore, let $\mathcal{X} = \mathcal{E} = \mathbb{R}^n$ and $\mathcal{C} = [\ell, u]_{\mathbb{Z}}$ with $\ell \leq 0 \leq u$. Let $B$ be the identity map, so that $\mathcal{D} = \mathcal{C}$. Furthermore, let $\mathcal{P}$ be the set of orthogonal projectors onto all coordinate subspaces $\mathcal{E}_S = \{y \in \mathbb{R}^n \,:\, y_i = 0 \,\forall\, i \notin S\}$ of $\mathbb{R}^n$, where $S \subseteq [n]$. For $P \in \mathcal{P}$, define its nonnegative weight as $\nu(P) := \operatorname{rank}(P)$, and define $\overline{P} := I_n - P$, where $I_n$ denotes the identity mapping on $\mathbb{R}^n$. Thus, if $P$ projects onto $\mathcal{E}_I$, then $\nu(P) = |I|$. Finally, let the norm $\|\cdot\|$ be the usual $\ell_1$-norm.

The general recovery problem (2.3) becomes

$$\min\{\|x\|_1 \,:\, Ax = b,\ x \in [\ell, u]_{\mathbb{Z}}\}, \tag{3.12}$$

which is a relaxation of

$$\min\{\|x\|_0 \,:\, Ax = b,\ x \in [\ell, u]_{\mathbb{Z}}\}. \tag{3.13}$$

In contrast to the classical case, where the $\ell_1$-minimization problem is the convex relaxation of the nonconvex $\ell_0$-minimization problem, (3.12) is nonconvex but can be formulated as a MIP. Furthermore, note that both (3.12) and (3.13) are $\mathcal{NP}$-hard problems [154].

Since sparse integral (box-constrained) vectors are a special case of sparse vectors, Assumptions (A1) to (A3) are satisfied in the integral case as well, except for the condition $c_1 + c_2 \in \mathcal{C}$ for all $c_1$, $c_2 \in \mathcal{C}$ in Assumption (A1). This condition may be violated due to the presence of box-constraints $\ell \leq x \leq u$. However, the following remark states that a simpler condition suffices for the statements in Chapter 2.

**Remark 3.16.** In the proofs of the recovery results in Chapter 2, i.e., Theorems 2.10 and 2.20 and Corollary 2.28, the assumption $c_1 + c_2 \in \mathcal{C}$ for all $c_1$, $c_2 \in \mathcal{C}$ is used. However, a closer inspection of these proofs reveals that a less restrictive assumption suffices. Indeed, in the proofs we only need the condition

$$PBc_1,\, \overline{P}Bc_2 \in \mathcal{D} \implies PBc_1 + \overline{P}Bc_2 \in \mathcal{D} \tag{3.14}$$

for all $c_1$, $c_2 \in \mathcal{C} + (-\mathcal{C})$ and all $P \in \mathcal{P}$.

Clearly, Condition (3.14) is satisfied for box-constrained integral vectors, so that the general framework from Chapter 2 is applicable. Next we prove that Assumption (A4) holds. To do so, note that for $v \in [\ell - u, u - \ell]_{\mathbb{Z}}$, the decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in [\ell, u]_{\mathbb{Z}}$ with $\|\hat{v}^{(2)}\|_1$ minimal is unique and given by

$$\hat{v}_i^{(1)} = \begin{cases} v_i, & \text{if } \ell_i \leq v_i \leq u_i, \\ u_i, & \text{if } v_i > u_i, \\ \ell_i, & \text{if } v_i < \ell_i, \end{cases} \quad \text{and} \quad \hat{v}_i^{(2)} = \begin{cases} 0, & \text{if } \ell_i \leq v_i \leq u_i, \\ u_i - v_i, & \text{if } v_i > u_i, \\ \ell_i - v_i, & \text{if } v_i < \ell_i, \end{cases} \tag{3.15}$$

for all $i \in [n]$.

**Lemma 3.17.** *Let $\ell \in (\mathbb{R} \cup \{-\infty\})^n$ and $u \in (\mathbb{R} \cup \{\infty\})^n$ with $\ell \leq 0 \leq u$. Let $P \in \mathcal{P}$ be the orthogonal projection onto $\mathcal{E}_S$ with $S \subseteq [n]$. Let $x$, $z \in [\ell, u]_{\mathbb{Z}}$ with $v := x - z$.*

*Furthermore, let* $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in [\ell, u]_{\mathbb{Z}}$ *be the minimal decomposition in* (3.15)*. Then,*

$$\|x\|_1 \leq \|z\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1.$$

*Proof.* For $x$, $z \in [\ell, u]_{\mathbb{Z}}$ and $v := x - z$, let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in [\ell, u]_{\mathbb{Z}}$ be the minimal decomposition in (3.15). This implies

$$\begin{aligned}
&\|z\|_1 - \|x\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1 \\
&= \|z_S\|_1 + \|z_{\overline{S}}\|_1 - \|x_S\|_1 + \|x_{\overline{S}}\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|x_{\overline{S}} - z_{\overline{S}}\|_1 \\
&\geq \|z_S\|_1 + \|z_{\overline{S}}\|_1 - \|x_S\|_1 + \|x_{\overline{S}}\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|x_{\overline{S}}\| - \|z_{\overline{S}}\|_1 \\
&= \|z_S\|_1 - \|x_S\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 \\
&= \sum_{i \in S} \left(|z_i| - |x_i| + |\hat{v}_i^{(1)}| - |\hat{v}_i^{(2)}|\right).
\end{aligned}$$

For $i \in S$ with $\ell_i \leq v_i \leq u_i$, it holds that $\hat{v}_i^{(1)} = v_i$ and $\hat{v}_i^{(2)} = 0$, which yields

$$|z_i| - |x_i| + |\hat{v}_i^{(1)}| - |\hat{v}_i^{(2)}| = |z_i| - |x_i| + |x_i - z_i| \geq 0.$$

For $i \in S$ with $v_i > u_i$, it holds that $\hat{v}_i^{(1)} = u_i$ and $\hat{v}_i^{(2)} = u_i - v_i \leq 0$. Thus,

$$|z_i| - |x_i| + |\hat{v}_i^{(1)}| - |\hat{v}_i^{(2)}| = |z_i| + z_i - (|x_i| + x_i) + |u_i| + u_i,$$

with $|z_i| + z_i \geq 0$ for $z_i \geq 0$ and $|z_i| + z_i = 0$ for $z_i < 0$. Since $u_i < v_i = x_i - z_i$ and $x_i \leq u_i$, we have $z_i < 0$. This yields

$$|z_i| + z_i - (|x_i| + x_i) + |u_i| + u_i \geq 2(u_i - x_i) \geq 0$$

for all $i \in S$ with $v_i > u_i$. The last case $v_i < \ell_i$ is completely analogous by noting that $v_i < \ell_i$ implies $z_i > 0$, which yields

$$|z_i| - |x_i| + |\hat{v}_i^{(1)}| - |\hat{v}_i^{(2)}| = |z_i| - z_i - (|x_i| - x_i) + |\ell_i| - \ell_i \geq 0.$$

This shows $|z_i| - |x_i| + |\hat{v}_i^{(1)}| - |\hat{v}_i^{(2)}| \geq 0$ for all $i \in S$, which implies

$$\|x\|_1 \leq \|z\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1. \qquad \square$$

Lemma 3.17 shows that Assumption (A4) is satisfied, so that by Theorem 2.10, the null space property (NSP$^{\mathcal{C}}$) characterizes uniform recovery for sparse box-constrained integral vectors. This NSP can be simplified as shown in the following theorem. This simplification is based on the idea to split a vector $x \in \mathbb{R}^n$ into its positive and negative part $x^+ \in \mathbb{R}^n_+$ and $x^- \in \mathbb{R}^n_+$ with $x = x^+ - x^-$, and to write $Ax = b$ as $(A, -A)(x^+, x^-) = b$.

**Theorem 3.18.** *Let $A \in \mathbb{R}^{m\times n}$ and $s \geq 0$ and define*

$$K := \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} u \\ -\ell \end{pmatrix} \right]_{\mathbb{Z}} \, : \, x_i \cdot y_i = 0, \; i \in [n] \right\}.$$

*Then every $s$-sparse $x^{(0)} \in [\ell, u]_{\mathbb{Z}}$ is the unique solution of* (3.12) *if and only if*

$$-\begin{pmatrix} v_{\overline{S}} \\ w_{\overline{S}} \end{pmatrix} \in K \quad \Longrightarrow \quad \sum_{i=1}^{n} \big( v_i + w_i \big) < 0 \tag{3.16}$$

*holds for all $(v^\top, w^\top)^\top \in \operatorname{null}(A, -A) \cap (K + (-K))$ with $(v^\top, w^\top)^\top \neq (0^\top, 0^\top)^\top$ and all index sets $S \subseteq [n]$, $|S| \leq s$.*

Before we prove this result, note that a similar result already appears in Lange et al. [154], but without the complementarity constraints in $K$. However, the following example shows that the complementarity constraints cannot be omitted.

**Example 3.19.** Let $\ell = (-1, -1, -1)^\top$, $u = (1, 1, 1)^\top$ and consider the matrix

$$A = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3},$$

which has a trivial null space $\operatorname{null}(A) = \{(0, 0, 0)^\top\}$. Consequently, $A$ has full rank, so that the system of linear equations $Ax = b$ has a unique solution for all $b \in \mathbb{R}^3$. Thus, every sparse $x^{(0)} \in \mathbb{Z}^3$ with $\ell \leq x^{(0)} \leq u$ is the unique optimal solution of the recovery problem $\min\{\|x\|_1 \, : \, Ax = Ax^{(0)}, \; \ell \leq x \leq u, \; x \in \mathbb{Z}^3\}$. The null space of the matrix $(A, -A)$ is given by $\operatorname{null}(A, -A) = \{(\alpha, \beta, \gamma, \alpha, \beta, \gamma)^\top \, : \, \alpha, \, \beta, \, \gamma \in \mathbb{R}\}$. Thus, for all $(v^\top, w^\top)^\top \in \operatorname{null}(A, -A) \cap (K + (-K))$ with $(v^\top, w^\top)^\top \neq (0^\top, 0^\top)^\top$, there exists no index $i$ with $v_i \cdot w_i = 0$, so that the NSP condition (3.16) trivially holds. Define

$$\tilde{K} := \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} u \\ -\ell \end{pmatrix} \right]_{\mathbb{Z}},$$

i.e., $\tilde{K}$ is the set $K$ without the complementarity constraint. Then,

$$(v^\top, w^\top)^\top = (1, -1, 0, 1, -1, 0)^\top \in \operatorname{null}(A, -A) \cap (\tilde{K} + (-\tilde{K})),$$

with $-(v_{\overline{S}}^\top, w_{\overline{S}}^\top)^\top \in \tilde{K}$ for $S = \{1\}$ and $\sum_{i=1}^n (v_i + w_i) = 0 \geq 0$, which is a contradiction to (3.16) with $\tilde{K}$ instead of $K$. This shows that the complementarity constraint is required in order to obtain a condition which is not only sufficient but also necessary for uniform recovery.

We now prove Theorem 3.18.

*Proof of Theorem 3.18.* Let $A \in \mathbb{R}^{m\times n}$, $s \geq 0$, and recall $\mathcal{C} = [\ell, u]_{\mathbb{Z}}$. We first show that (3.16) is equivalent to (NSP$^{\mathcal{C}}$). Then, Theorem 2.10 yields the desired equivalence between uniform recovery and (3.16).

Therefore, assume first that (NSP$^{\mathcal{C}}$) holds and let $S \subseteq [n]$, $|S| \leq s$. Let $(v^\top, w^\top)^\top \in \operatorname{null}(A, -A) \cap (K + (-K))$ with $(v^\top, w^\top)^\top \neq (0^\top, 0^\top)^\top$, and $-(v_{\overline{S}}^\top, w_{\overline{S}}^\top)^\top \in K$. Thus, $v - w \in \operatorname{null}(A)$, and $-v_{\overline{S}} \in [0, u]_{\mathbb{Z}}$ as well as $w_{\overline{S}} \in [\ell, 0]_{\mathbb{Z}}$. This implies $-(v - w)_{\overline{S}} \in [\ell, u]_{\mathbb{Z}}$. Since $(v^\top, w^\top)^\top \in K + (-K)$, there exist $((x^{(1)})^\top, (x^{(2)})^\top)^\top$, $((y^{(1)})^\top, (y^{(2)})^\top)^\top \in K$ with $v = x^{(1)} - y^{(1)}$ as well as $w = x^{(2)} - y^{(2)}$, so that it is easy to see that $v \neq w$. Indeed, assume there exists $j \in [n]$ with $v_j = w_j \neq 0$. If $x_j^{(1)} \neq 0$, then $x_j^{(2)} = 0$, $y_j^{(2)} \neq 0$ and $y_j^{(1)} = 0$ due to the complementarity constraints in $K$ and $x_j^{(1)} - y_j^{(1)} = x_j^{(2)} - y_j^{(2)}$. Thus, $x_j^{(1)} = -y_j^{(2)}$ and $v_j = x_j^{(1)} > 0$ as well as $w_j = -y_j^{(2)} < 0$. This is a contradiction to $v_j = w_j \neq 0$. The case $x_j^{(1)} = 0$ is completely analogous. Define $f := v - w$. Then,

$$f = v - w = \big(\underbrace{x^{(1)} - x^{(2)}}_{\in[\ell,u]_{\mathbb{Z}}}\big) - \big(\underbrace{y^{(1)} - y^{(2)}}_{\in[\ell,u]_{\mathbb{Z}}}\big) \in \operatorname{null}(A) \cap [\ell - u, u - \ell]_{\mathbb{Z}},$$

since $((x^{(1)})^\top, (x^{(2)})^\top)^\top \in K$ and $((y^{(1)})^\top, (y^{(2)})^\top)^\top \in K$. Moreover, we have $-f_{\overline{S}} = w_{\overline{S}} - v_{\overline{S}} \in [\ell, u]_{\mathbb{Z}}$, since $-(v_{\overline{S}}^\top, w_{\overline{S}}^\top)^\top \in K$ as well, so that we can apply (NSP$^{\mathcal{C}}$). Let $\hat{f}^{(1)}$, $\hat{f}^{(2)} \in [\ell, u]_{\mathbb{Z}}$ with $f = \hat{f}^{(1)} - \hat{f}^{(2)}$ and $\|\hat{f}^{(2)}\|_1$ minimal. Then,

$$\|\hat{f}_S^{(1)}\|_1 - \|\hat{f}_S^{(2)}\|_1 < \|f_{\overline{S}}\|_1. \tag{3.17}$$

Additionally, we define

$$x := \begin{cases} 0, & \text{if } j \in \overline{S}, \\ x_j^{(1)} - x_j^{(2)}, & \text{if } j \in S, \end{cases} \quad \text{and} \quad z := \begin{cases} -f_j, & \text{if } j \in \overline{S}, \\ y_j^{(1)} - y_j^{(2)}, & \text{if } j \in S, \end{cases}$$

so that $f = x - z$ and $x$, $z \in [\ell, u]_{\mathbb{Z}} = \mathcal{C}$. Assumption (A4) for $f$, $x$, $z$ and $\hat{f}^{(1)}$, $\hat{f}^{(2)}$ yields

$$0 \leq \|z\|_1 - \|x\|_1 + \|\hat{f}_S^{(1)}\|_1 - \|\hat{f}_S^{(2)}\|_1 - \|f_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1. \tag{3.18}$$

Combining (3.17) and (3.18) shows

$$\begin{aligned}0 < \|z\|_1 - \|x\|_1 + 2\|x_{\overline{S}}\|_1 &= \|z_S\|_1 - \|x_S\|_1 + \|z_{\overline{S}}\|_1 + \|x_{\overline{S}}\|_1 \\ &= \sum_{i \in S} \Big( |y_i^{(1)} - y_i^{(2)}| - |x_i^{(1)} - x_i^{(2)}| \Big) + \sum_{i \in \overline{S}} |w_i - v_i|.\end{aligned}$$

Since $((y^{(1)})^\top, (y^{(2)})^\top)^\top$, $((x^{(1)})^\top, (x^{(2)})^\top)^\top \in K$, and $-(v_{\overline{S}}^\top, w_{\overline{S}}^\top)^\top \in K$, we obtain

$$0 < \sum_{i \in S} \Big( y_i^{(1)} + y_i^{(2)} - \big(x_i^{(1)} + x_i^{(2)}\big) \Big) + \sum_{i \in \overline{S}} -(v_i + w_i) = -\sum_{i=1}^{n} (v_i + w_i),$$

which shows that $\sum_{i=1}^n (v_i + w_i) < 0$, as desired in (3.16).

For the reverse implication assume that (3.16) holds. Let $S \subseteq [n]$, $|S| \leq s$ and let $v \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ with $v \neq 0$ as well as $-v_{\overline{S}} \in \mathcal{C}$. Furthermore, let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ and $\|\hat{v}^{(2)}\|_1$ minimal. Define

$$x \coloneqq (\hat{v}_S^{(1)})^+ - (\hat{v}_S^{(2)})^+ - (-v_{\overline{S}})^+, \quad y \coloneqq (\hat{v}_S^{(1)})^- - (\hat{v}_S^{(2)})^- - (-v_{\overline{S}})^-.$$

Then, $(A, -A)(x^\top, y^\top)^\top = Av = 0$. Furthermore, we have $(x^\top, y^\top)^\top \neq (0^\top, 0^\top)^\top$ since $v \neq 0$ and $(x^\top, y^\top)^\top \in K + (-K)$ with $-(x_{\overline{S}}^\top, y_{\overline{S}}^\top)^\top \in K$. Thus, (3.16) implies

$$\begin{aligned}0 > \mathbb{1}^\top \begin{pmatrix} x \\ y \end{pmatrix} &= \sum_{i \in S} (x_i + y_i) + \sum_{i \in \overline{S}} (x_i + y_i) \\ &= \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\| - \sum_{i \in \overline{S}} \big((-v_i)^+ + (-v_i)^-\big) \\ &= \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\| - \|v_{\overline{S}}\|_1,\end{aligned}$$

which shows that (NSP$^{\mathcal{C}}$) is satisfied. This concludes the proof. □

The complementarity constraints $x_i \cdot y_i = 0$ in $K$ are due to the split into a positive and a negative part. This already shows that the introduction of bounds leads to different recovery conditions, in contrast to the situation of classical sparse recovery over $\mathbb{R}^n$. For testing the NSP in Theorem 3.18, one needs to take care of the complementarity constraints $x_i \cdot y_i = 0$. This can be done by, e.g., using methods by Fischer and Pfetsch [95, 96].

**Remark 3.20.** It is also possible to use the exact recovery problem (3.13) instead of (3.12) for recovery of sparse integral vectors. If there are finite bounds, then Problem (3.13) can also be formulated as a MIP by using binary variables to model the nonconvex $\ell_0$-term in the objective function. Recall that both (3.12) and (3.13) are $\mathcal{NP}$-hard problems, see [154]. Recovery conditions for integral sparse recovery with and without bounds when solving (3.13) can be found in [154]. In the classical case of sparse recovery, the recovery condition for $\ell_0$-minimization is $\operatorname{spark}(A) > 2s$, where $\operatorname{spark}(A)$ denotes the smallest number of linear dependent columns in $A$, see Remark 2.16.

**Without Box-Constraints** If there are no box-constraints, that is, $\mathcal{C} = \mathbb{Z}^n$, then Condition (3.16) can be further simplified to

$$\|v_S\|_1 < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \big(\operatorname{null}(A) \cap \mathbb{Z}^n\big) \setminus \{0\},\ S \subseteq [n],\ |S| \leq s, \tag{3.19}$$

which is exactly the classical NSP for the recovery of sparse vectors restricted to integral vectors in the null space of $A$, see Example (2.12.1). Indeed, for $\mathcal{C} = \mathbb{Z}^n$ and $v \in \mathbb{Z}^n$, the decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\hat{v}^{(1)}, \hat{v}^{(2)} \in \mathbb{Z}^n$ and $\|\hat{v}^{(2)}\|_1$ minimal is unique and given by $\hat{v}^{(1)} = v$ and $\hat{v}^{(2)} = 0$. Consequently, (NSP$^{\mathcal{C}}$) simplifies to (3.19). The NSP (3.19) already appears in [154]. This shows that the split into positive and negative part together with the complementarity constraints in $K$ are not necessary if no box-constraints are present. Of course, the NSP (3.16) with $\ell_i = -\infty$ and $u_i = \infty$ for all $i \in [n]$ is also an equivalent characterization of uniform recovery of sparse integral vectors. By removing the integrality condition, we further obtain an NSP for sparse vectors, which is of course equivalent to the classical condition (NSP).

Clearly, for rational matrices $A \in \mathbb{Q}^{m\times n}$, vectors in the null space of $A$ can always be scaled to be integral, so that there is no difference between the recovery of integral and general $x$, see also [154]. In the presence of additional box-constraints on some entries of $x$, this is no longer true. The condition

$$\|v_S\|_1 < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \big(\operatorname{null}(A) \cap [\ell - u, u - \ell]_{\mathbb{Z}}\big) \setminus \{0\},\ S \subseteq [n],\ |S| \leq s,$$

is shown to be only sufficient but not necessary for uniform recovery of $x \in [\ell, u]_{\mathbb{Z}}$ using (3.12) in [154]. This is in contrast to the general case without integrality. There, additional bounds do not influence the recovery conditions since vectors in the null space of $A$ can always be scaled accordingly.

**With Nonnegativity** If the box-constraints are given by $\ell_i = 0$ and $u_i = \infty$ for all $i \in [n]$, i.e., $x \geq 0$ componentwise and $\mathcal{C} = \mathbb{Z}^n_+$, then Example (2.12.2) shows that Assumption (A4) is satisfied. Thus, uniform recovery is characterized by (NSP$^{\mathcal{C}}$), which simplifies to

$$v_{\overline{S}} \leq 0 \implies \mathbb{1}^\top v < 0 \quad \forall\, v \in \big(\operatorname{null}(A) \cap \mathbb{Z}^n\big) \setminus \{0\},\ S \subseteq [n],\ |S| \leq s, \tag{3.20}$$

analogously to Example (2.12.2). The NSP (3.20) also appears in [154]. As in the case without bounds, this null space property shows that exploiting integrality for rational matrices $A \in \mathbb{Q}^{m\times n}$ does not lead to improved recovery conditions. If additional upper bounds are introduced, i.e., $\mathcal{C} = [0, u]_{\mathbb{Z}}$, then the variable split in Theorem 3.18 is not needed, and it can be shown that Condition (3.20) for all $v \in (\mathrm{null}(A) \cap [-u, u]_{\mathbb{Z}}) \setminus \{0\}$ and all $S \subseteq [n]$ with $|S| \leq s$ characterizes uniform recovery, see [154].

**Individual Recovery** In the case of individual recovery of a fixed sparse integral $x^{(0)} \in [\ell, u]_{\mathbb{Z}}$ with $\ell \leq 0 \leq u$, the results which can be obtained from Section 2.4 become exactly the results in [154], so that we do not repeat them here. In [154], it is shown that the direct adaption of the corresponding statements in the classical (non-integral) case (see (2.27) and (2.28)) to integrality yields a characterization, if an additional nonnegativity constraint is present. However, without the additional nonnegativity, the resulting conditions are only sufficient for individual recovery of integral vectors, in contrast to the classical case. It is possible to obtain a simple characterization of individual integral recovery by using Lemma 2.34, which reads

$$\|x^{(0)} + v\|_1 > \|x^{(0)}\|_1 \quad \forall\, v \in (\mathrm{null}(A) \cap \mathbb{Z}^n) \setminus \{0\}.$$

In the presence of bounds, a variable split as in Theorem 3.18 can be used to obtain the following characterization of individual recovery of integral vectors in the presence of bounds, which resembles the usual null space properties:

$$\begin{pmatrix} v \\ w \end{pmatrix} + \begin{pmatrix} x_+^{(0)} \\ x_-^{(0)} \end{pmatrix} \in [0, \left(\begin{smallmatrix} u \\ -\ell \end{smallmatrix}\right)]_{\mathbb{Z}} \implies \mathbb{1}^\top \begin{pmatrix} v \\ w \end{pmatrix} > 0$$

for all $(v^\top, w^\top)^\top \in \mathrm{null}(A, -A) \cap [\left(\begin{smallmatrix} -u \\ \ell \end{smallmatrix}\right), \left(\begin{smallmatrix} u \\ -\ell \end{smallmatrix}\right)]_{\mathbb{Z}} \setminus \{0\}$, see [154, Theorem 4.22]. Note that in contrast to uniform recovery in Theorem 3.18, this NSP for individual recovery does not need complementarity constraints.

**Stability and Robustness** Let us lastly consider stable and robust recovery of sparse integral vectors. To do so, let $|||\cdot|||$ be a norm on $\mathbb{R}^m$ in which the recovery error shall be measured, e.g., the $\ell_1$- or the $\ell_2$-norm. Without additional bounds and with an additional nonnegativity constraint, the corresponding NSPs can be directly obtained from the NSP for stable and robust (nonnegative) recovery in Section 2.3.3 by demanding that the respective condition only needs to hold for integral vectors in the null space of $A$. Using $\mathcal{C} = \mathbb{Z}^n$ in Theorem 2.20 yields a characterization for robust integral recovery and Theorem 2.23 presents the corresponding error bound. The results for stable integral recovery can be obtained from

Corollary 2.28 and Corollary 2.29, respectively. Furthermore, replacing $\mathcal{C} = \mathbb{Z}^n$ by $\mathcal{C} = \mathbb{Z}^n_+$ yields the corresponding results for integral nonnegative vectors. In both cases, Assumption (A5) is satisfied analogously to the case of sparse (nonnegative) vectors without integrality, c.f. Section 2.3.3. Moreover, $\|v\|_1 = \|v_S\|_1 + \|v_{\overline{S}}\|_1$ clearly holds for all $v \in \mathbb{R}^n$ and all $S \subseteq [n]$.

In the presence of box-constraints $x \in \mathcal{C} = [\ell, u]_{\mathbb{Z}}$ with $\ell \leq 0 \leq u$, $\ell \in (\mathbb{R} \cup \{-\infty\})^n$ and $u \in (\mathbb{R} \cup \{\infty\})^n$, Assumption (A5) does not hold due to the structure of the minimal decomposition of $v \in \mathcal{C} + (-\mathcal{C})$ in (3.15). However, this problem can be avoided by using a variable split. The robust recovery problem

$$\min \{\|x\|_1 \, : \, |||Ax - b||| \leq \eta, \; x \in [\ell, u]_{\mathbb{Z}}\}$$

is equivalent to the recovery problem

$$\min \left\{ \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_1 \, : \, \left|\left|\left| (A, -A) \begin{pmatrix} x \\ y \end{pmatrix} - b \right|\right|\right| \leq \eta, \; \begin{pmatrix} x \\ y \end{pmatrix} \in K \right\}, \tag{3.21}$$

where the set $K$ is defined as in Theorem 3.18, i.e.,

$$K := \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} u \\ -\ell \end{pmatrix} \right]_{\mathbb{Z}} \, : \, x_i \cdot y_i = 0, \; i \in [n] \right\}.$$

In order to formulate the setting after using the variable split in the general framework from Chapter 2, we use $\mathcal{X} = \mathcal{E} = \mathbb{R}^{2n}$ and $\mathcal{C} = K$. Furthermore, $B$ is the identity map, so that $\mathcal{D} = \mathcal{C}$, and $\mathcal{P}$ is the set of orthogonal projectors onto all coordinate subspaces $\mathcal{E}_S = \{(x^\top, y^\top)^\top \in \mathbb{R}^{2n} \, : \, x_i = y_i = 0 \; \forall \, i \notin S\}$ of $\mathbb{R}^{2n}$, where $S \subseteq [n]$. For $P \in \mathcal{P}$, define its nonnegative weight as $\nu(P) := \operatorname{rank}(P)/2$, and define $\overline{P} := I_{2n} - P$, where $I_{2n}$ denotes the identity mapping on $\mathbb{R}^{2n}$. Finally, let the norm $\|\cdot\|$ be the $\ell_1$-norm. Then, the general robust recovery problem (2.9) becomes the recovery problem (3.21). The general stable null space property (sNSP$^{\mathcal{C}}_\rho$) for $(A, -A) \in \mathbb{R}^{m \times 2n}$ reads

$$-\begin{pmatrix} v_{\overline{S}} \\ w_{\overline{S}} \end{pmatrix} \in K \implies \sum_{i \in S} (v_i + w_i) \leq \rho \left\| \begin{pmatrix} v_{\overline{S}} \\ w_{\overline{S}} \end{pmatrix} \right\|_1 \tag{3.22}$$

for all $(v^\top, w^\top)^\top \in \operatorname{null}(A, -A) \cap (K + (-K))$ and all $S \subseteq [n]$ with $|S| \leq s$. Similarly, the general robust null space property (rNSP$^{\mathcal{C}}_{\rho,\tau}$) for $(A, -A) \in \mathbb{R}^{m \times 2n}$ becomes

$$-\begin{pmatrix} v_{\overline{S}} \\ w_{\overline{S}} \end{pmatrix} \in K \implies \sum_{i \in S} (v_i + w_i) \leq \rho \left\| \begin{pmatrix} v_{\overline{S}} \\ w_{\overline{S}} \end{pmatrix} \right\|_1 + \tau \left|\left|\left| (A, -A) \begin{pmatrix} v \\ w \end{pmatrix} \right|\right|\right| \tag{3.23}$$

for all $(v^\top, w^\top)^\top \in K + (-K)$ and all $S \subseteq [n]$ with $|S| \leq s$.

In this setting, Assumptions (A1) to (A5) are satisfied, analogously to the case of sparse nonnegative vectors without integrality, see Example (2.12.2). Thus, Corollary 2.28 and Theorem 2.20 provide the following characterizations of stable and robust recovery using (3.22) and (3.23), respectively.

**Lemma 3.21.** *Let $A \in \mathbb{R}^{m\times n}$ and $s \geq 0$. Then, the following statements hold, where for $x = (\alpha^\top, \beta^\top)^\top \in \mathbb{R}^{2n}$ and $S \subseteq [n]$, we write $x_S = (\alpha_S^\top, \beta_S^\top)^\top$.*

1. *The matrix $(A, -A)$ satisfies the integral stable NSP* (3.22) *of order $s$ with constant $\rho \in (0,1)$ if and only if*

$$-v_{\overline{S}} \in K \implies \|x - z\|_1 \leq \tfrac{1+\rho}{1-\rho}\big(\|z\|_1 - \|x\|_1 + 2\|x_{\overline{S}}\|_1\big)$$

*holds for all $x$, $z \in K$ with $(A,-A)x = (A,-A)z$, $v := x - z$ and all $S \subseteq [n]$ with $|S| \leq s$.*

2. *The matrix $(A, -A)$ satisfies the integral robust NSP* (3.23) *of order $s$ with constants $\rho \in (0,1)$ and $\tau > 0$ if and only if*

$$\begin{aligned}-v_{\overline{S}} \in K \implies \|x - z\|_1 \leq \tfrac{1+\rho}{1-\rho}\big(\|z\|_1 - \|x\|_1 + 2\|x_{\overline{S}}\|_1\big)& \\ + \tfrac{2\tau}{1-\rho}\|\!|(A,-A)(x-z)|\!\|&\end{aligned}$$

*holds for all $x$, $z \in K$, $v := x - z$ and all $S \subseteq [n]$ with $|S| \leq s$.*

*Proof.* Let $\mathcal{C} = K$. The first part directly follows from Corollary 2.28 and the second part is due to Theorem 2.20 by using the variable split as outlined above. □

The corresponding error bounds for stable and robust recovery of sparse box-constrained integral vectors can be obtained from Corollary 2.29 and Theorem 2.23 with $\mathcal{C} = K$, respectively. These error bounds hold for the recovery problems after using the variable split. In this setting, the error $\sigma_s$ of the best $s$-term approximation of $x = ((x^{(1)})^\top, (x^{(2)})^\top)^\top \in \mathbb{R}^{2n}$ in Definition 2.22 reads

$$\sigma_s(x) = \min\left\{\left\|\begin{pmatrix}x^{(1)}\\x^{(2)}\end{pmatrix} - \begin{pmatrix}z^{(1)}\\z^{(2)}\end{pmatrix}\right\|_1 \;:\; \exists\, S \subseteq [n],\ |S| \leq s,\ \text{with}\ \begin{pmatrix}z^{(1)}_{\overline{S}}\\z^{(2)}_{\overline{S}}\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}\right\},$$

and any $z = ((z^{(1)})^\top, (z^{(2)})^\top)^\top \in \mathbb{R}^{2n}$ attaining this minimum is called a best $s$-term approximation of $x$.

**Lemma 3.22.** *Let $A \in \mathbb{R}^{m\times n}$ and $s \geq 0$. Then, the following statements hold, where for $x = (\alpha^\top, \beta^\top)^\top \in \mathbb{R}^{2n}$ and $S \subseteq [n]$, we again write $x_S = (\alpha_S^\top, \beta_S^\top)^\top$.*

1. *Let* $x^{(0)} \in K$ *and let* $\tilde{x}$ *be a solution of*

$$\min\{\|x\|_1 \,:\, (A,-A)x = (A,-A)x^{(0)},\ x \in K\}.$$

*Furthermore, let* $S \subseteq [n]$ *such that* $x^{(0)}_S$ *is a best* $s$*-term approximation of* $x^{(0)}$*. If the matrix* $(A,-A)$ *satisfies the integral stable NSP* (3.22) *of order* $s$ *with constant* $\rho \in (0,1)$ *and if* $-(x^{(0)}_{\overline{S}} - \tilde{x}_{\overline{S}}) \in K$*, then* $\tilde{x}$ *approximates* $x^{(0)}$ *with error*

$$\|x^{(0)} - \tilde{x}\|_1 \leq 2\tfrac{1+\rho}{1-\rho}\sigma_s(x^{(0)}).$$

2. *Let* $x^{(0)} \in K$ *and let* $\tilde{x}$ *be a solution of*

$$\min\{\|x\|_1 \,:\, |||(A,-A)x - b||| \leq \eta,\ x \in K\}$$

*with* $b = (A,-A)x^{(0)} + e$ *and* $|||e||| \leq \eta$*. Furthermore, let* $S \subseteq [n]$ *such that* $x^{(0)}_S$ *is a best* $s$*-term approximation of* $x^{(0)}$*. If the matrix* $(A,-A)$ *satisfies the integral robust NSP* (3.23) *of order* $s$ *with constants* $\rho \in (0,1)$ *and* $\tau > 0$ *and if* $-(x^{(0)}_{\overline{S}} - \tilde{x}_{\overline{S}}) \in K$*, then* $\tilde{x}$ *approximates* $x^{(0)}$ *with error*

$$\|x^{(0)} - \tilde{x}\|_1 \leq 2\tfrac{1+\rho}{1-\rho}\sigma_s(x^{(0)}) + \tfrac{4\tau}{1-\rho}\eta.$$

*Proof.* Let $\mathcal{C} = K$. The first part directly follows from Corollary 2.29, and the second statement is due to Theorem 2.23. □

If $x^{(0)} \in \mathbb{R}^{2n}$ is $s$-sparse, that is, there exists $S \subseteq [n]$ with $|S| \leq s$ and $x^{(0)}_{\overline{S}} = 0$, then $\sigma_s(x^{(0)}) = 0$ and the error bound in the second part of Lemma 3.22 becomes

$$\|x^{(0)} - \tilde{x}\|_1 \leq \tfrac{4\tau}{1-\rho}\eta.$$

Moreover, if the measurement error (or the noise level, respectively) satisfies $\eta = 0$, that is, the measurements are exact, then Lemma 3.22 asserts that $x^{(0)}$ is exactly recovered. Thus, we recover the statement from Theorem 3.18 about exact uniform recovery.

## 3.3 Constant Modulus Constraints on Vectors

Until now, all considered special cases of the general framework presented in Chapter 2 were settings and side constraints over the real numbers. At least the cases without additional side constraints such as nonnegativity or positive semidefiniteness can directly be extended to the complex setting. For instance, it is well known that the NSPs which emerge from the general framework in case of sparse complex vectors and sparse real vectors are in fact equivalent for a real linear sensing map $A$, see Foucart and Gribonval [102]. However, since for a complex number $x \in \mathbb{C}$, nonnegativity is not well-defined, the results for recovery of sparse (real) nonnegative vectors do not carry over to the complex setting. In this section, we explicitly consider the setting of complex vectors together with a side constraint that is more interesting for complex vectors than for real vectors. Namely, we demand that all entries $x_j \in \mathbb{C}$ of the vector $x \in \mathbb{C}^n$ to be recovered have a *constant modulus*, that is,

$$|x_j| = \sqrt{\mathrm{Re}[x_j]^2 + \mathrm{Im}[x_j]^2} \in \{0, c\}$$

holds for all $j \in [n]$ and for some $c \in \mathbb{R}$, where $\mathrm{Re}[x]$ and $\mathrm{Im}[x]$ denote the real and imaginary part of $x \in \mathbb{C}$, respectively. Clearly, if $|x_j| = 0$, then also $x_j = 0$, so that sparsity counts the number of entries with $|x_j| = 1$, that is, $\mathrm{Re}[x_j]$ or $\mathrm{Im}[x_j]$ are nonzero. Note that throughout this section, $i$ denotes the imaginary unit and $j$ is used for indices.

The assumption of constant modulus is frequently encountered in communication applications, see van der Veen and Paulraj [243] and the references therein. Typically, in order to transmit communication signals, a modulation is used, which varies properties of the signal such as amplitude, phase or frequency. Examples for modulation schemes include frequency modulation (FM) and phase modulation (PM) of analog signals, where either the frequency or the phase is varied, whereas the other properties remain constant. Modulation schemes for digital signals include frequency shift keying (FSK), phase shift keying (PSK), or minimum shift keying (MSK), where a finite alphabet of frequencies or phases are used to represent the signal. The resulting signals under such a modulation then have a constant modulus, which can be exploited in the reconstruction. Furthermore, the constant modulus property can also be exploited in the context of direction-of-arrival estimation or parameter estimation [155].

In the “EXPRESS” project within the SPP 1798, constant modulus was considered as one specific structure in the problem of joint antenna selection and phase-only beamforming in transmission networks [97]. In directional signal transmission via beamforming, radio frequency (RF) phase shifters are employed to vary the phase of the signal at the transmitters. In large networks, such as massive multiple-input/multiple-output (MIMO) systems, it is no longer affordable to connect each antenna element to a dedicated RF phase shifter for it to be able to transmit signals. Rather, using switches, there only exists a reduced number of costly RF phase shifters, which are connected to a subset of the inexpensive antenna elements. In hybrid massive MIMO systems, the RF phase shifters are based on analog beamformers, which require a fixed magnitude of the signals to be transmitted, since only

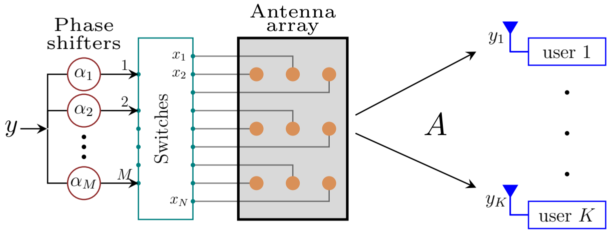


**Figure 3.1.** Schematic model of the system model for joint antenna selection and phase-only beamforming, taken from [97].

the phase of the signals can be varied. More specifically, consider a sensor array with $N$ antenna elements, $M$ phase shifters with $M \ll N$ and $K$ (single-antenna) users that need to be served. Let $x \in \mathbb{C}^N$ be the transmitted signal vector, and let $\alpha = (\alpha_1, \dots, \alpha_M)^\top \in \mathbb{C}^M$ be the set of $M$ analog beamformers, where $\alpha_m$ is the value of the $m$-th phase shifter with $|\alpha_m| = c$ for a constant $c \in \mathbb{R}$ and all $m \in [M]$. Throughout this section, we assume without loss of generality $c = 1$. Further, let $A \in \mathbb{C}^{N \times K}$ be the channel matrix with columns $a_1, \dots, a_K \in \mathbb{C}^N$. If the $n$-th antenna is connected to the $m$-th phase shifter, then $x_n = \alpha_m$, and $x_n = 0$ otherwise. The desired output at the $K$ users is given by $y \in \mathbb{C}^K$, and the actual output at the users can be expressed as $\hat{y} = A^\top x + e$, where $e$ represents complex i.i.d. additive white Gaussian noise. The underlying model is depicted in Figure 3.1.

In order to minimize hardware costs, the approach in [97] minimizes the number of required phase shifters by jointly assigning appropriate antenna elements to the phase shifters and designing the optimal phase values, while keeping the root-mean-square error between the desired and the actual output at the users below a given threshold $\sqrt{\delta}$. This problem can be formulated as

$$\begin{aligned} \min_{x \in \mathbb{C}^N} \ & \|x\|_0 \\ \text{s.t. } & \|y - A^\top x\|_2 \leq \sqrt{\delta}, \\ & |x_j| \in \{0,1\} \quad \forall j \in [N]. \end{aligned} \tag{3.24}$$

The rest of this section is structured as follows. In Section 3.3.1, we first show that the side constraint $|x_j| \in \{0, 1\}$ fits into our framework in Chapter 2. Then, Section 3.3.2 describes an algorithmic approach to solve Problem (3.24), together with a (primal) heuristic, which can be used in a solution process to obtain good feasible solutions. Lastly, Section 3.3.3 presents numerical experiments for the problem of joint antenna selection and phase-only beamforming, which we described above.

The contents of Sections 3.3.2 and 3.3.3 are taken from the publication [97], whereas Section 3.3.1 has been developed independently.

### 3.3.1 Constant Modulus Constraints in the General Framework

In order to derive the setting of constant modulus constraints from the general framework in Chapter 2, let $\mathcal{X} = \mathbb{C}^n$, $\mathcal{C} = \{x \in \mathbb{C}^n : |x_j| \in \{0,1\},\, j \in [n]\}$ and let $B$ be the identity map, so that $\mathcal{E} = \mathcal{X} = \mathbb{C}^n$ and $\mathcal{D} = \mathcal{C}$. We use a complex linear sensing map $A\colon \mathbb{C}^n \mapsto \mathbb{C}^m$ and the $\ell_1$-norm $\|\cdot\|_1$. The set $\mathcal{P}$ consists of projections $P_S$ onto subspaces of the form $\{x \in \mathbb{C}^n : \mathrm{Re}[x_j] = \mathrm{Im}[x_j] = 0,\, j \notin S\}$ with $S \subseteq [n]$. For a projection $P_S \in \mathcal{P}$ and $x \in \mathbb{C}^n$, we define $x_S \coloneqq P_S x$, i.e.,

$$\mathrm{Re}[(x_S)_j] = \begin{cases} \mathrm{Re}[x_j], & \text{if } j \in S, \\ 0, & \text{if } j \notin S, \end{cases} \qquad \mathrm{Im}[(x_S)_j] = \begin{cases} \mathrm{Im}[x_j], & \text{if } j \in S, \\ 0, & \text{if } j \notin S. \end{cases}$$

The nonnegative weight $\nu(P_S)$ of a projector $P_S \in \mathcal{P}$ is given by $\nu(P_S) = |S|$ and the associated map $\overline{P} = P_{\overline{S}}$. Note that we adopted the notation used in Chapter 2, in order to derive the setting of constant modulus constraints independent of the specific application explained above. Analogously to the case of box-constrained integral vectors, the condition $c_1 + c_2 \in \mathcal{C}$ for all $c_1$, $c_2 \in \mathcal{C}$ from Assumption (A1) does not hold, whereas the weaker condition

$$PBc_1,\, \overline{P}Bc_2 \in \mathcal{D} \implies PBc_1 + \overline{P}Bc_2 \in \mathcal{D}$$

for all $c_1$, $c_2 \in \mathcal{C} + (-\mathcal{C})$ holds. As stated in Remark 3.16, this condition suffices for the proofs of the statements in Chapter 2. The other conditions from Assumptions (A1) to (A3) are clearly satisfied. In order to show that Assumption (A4) holds as well, we need to find all decompositions of $v \in \mathcal{C} + (-\mathcal{C})$ into $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ and $\|\hat{v}^{(2)}\|_1$ minimal. The following lemma states a key property satisfied by all minimal decompositions, which allows to prove Assumption (A4).

**Lemma 3.23.** *Let $x$, $z \in \mathcal{C}$ and define $v \coloneqq x - z \in \mathcal{C} + (-\mathcal{C})$. Then, if $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ is a decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\|\hat{v}^{(2)}\|_1$ minimal, then the inequality*

$$|z_j| - |x_j| + |\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}| \geq 0$$

*holds for all $j \in [n]$.*

*Proof.* Let $x$, $z \in \mathcal{C}$ and define $v \coloneqq x - z \in \mathcal{C} + (-\mathcal{C})$. Furthermore, let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ be a decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\|\hat{v}^{(2)}\|_1$ minimal. Let $j \in [n]$. We distinguish between two cases.

1. If $v_j = 0$, then $x_j = z_j$. In this case, the minimality of $\|\hat{v}^{(2)}\|_1$ clearly implies $\hat{v}_j^{(1)} = \hat{v}_j^{(2)} = 0$, and since $x$, $z \in \mathcal{C}$, we have $|z_j| - |x_j| + |\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}| = 0$.
2. If $v_j \neq 0$, there are three subcases: Exactly one of $x_j$, $z_j$ can be nonzero, or both can be nonzero. For the first two cases, the minimality of $\|\hat{v}^{(2)}\|_1$ implies

$$\hat{v}_j^{(1)} = \begin{cases} x_j, & \text{if } x_j \neq 0 \text{ and } z_j = 0, \\ -z_j, & \text{if } x_j = 0 \text{ and } z_j \neq 0, \end{cases} \qquad \text{and} \qquad \hat{v}_j^{(2)} = 0,$$

which yields $|z_j| - |x_j| + |\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}| \geq 0$ as well. For the remaining case $x_j \neq 0$ and $z_j \neq 0$, the corresponding minimal decomposition cannot be explicitly stated. However, since $x$, $z$, $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$, we have $|z_j| - |x_j| = 0$ and $|\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}| \geq 0$. Indeed, $|\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}| < 0$ implies $\hat{v}_j^{(1)} = 0$ and $\hat{v}_j^{(2)} \neq 0$, since $\hat{v}_j^{(1)}$, $\hat{v}_j^{(2)} \in \mathcal{C}$. Thus, switching $\hat{v}_j^{(1)}$ and $\hat{v}_j^{(2)}$ (as well as their signs) leads to a smaller $\ell_1$-norm of $\hat{v}^{(2)}$, which is a contradiction to the minimality of $\|\hat{v}^{(2)}\|_1$. Consequently, also in this case, the inequality $|z_j| - |x_j| + |\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}| \geq 0$ holds. □

Using Lemma 3.23 we can now show that Assumption (A4) is satisfied. The argument is similar to the proof of Lemma 3.17.

**Lemma 3.24.** *Let $x$, $z \in \mathcal{C} = \{x \in \mathbb{C}^n : |x_j| \in \{0,1\}, j \in [n]\}$. Let $S \subseteq [n]$. Furthermore, define $v := x - z$ and let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ be a minimal decomposition. Then,*

$$\|x\|_1 \leq \|z\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1.$$

*Proof.* Let $x$, $z \in \mathcal{C} = \{x \in \mathbb{C}^n : |x_j| \in \{0,1\}, j \in [n]\}$ and define $v := x - z$. Let $S \subseteq [n]$. Furthermore let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ be a decomposition $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ with $\|\hat{v}^{(2)}\|_1$ minimal. Then,

$$\begin{aligned} &\|z\|_1 - \|x\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|x_{\overline{S}}\|_1 \\ &= \sum_{j \in S} \Big(|z_j| - |x_j| + |\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}|\Big) + \sum_{j \notin S} \Big(|z_j| - |x_j| - |v_j| + 2|x_j|\Big) \\ &= \sum_{j \in S} \Big(|z_j| - |x_j| + |\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}|\Big) + \sum_{j \notin S} \Big(|z_j| + |x_j| - |x_j - z_j|\Big) \\ &\geq 0, \end{aligned}$$

since $|x_j - z_j| \leq |z_j| + |x_j|$ and $|z_j| - |x_j| + |\hat{v}_j^{(1)}| - |\hat{v}_j^{(2)}| \geq 0$ for all $j \in S$ by Lemma 3.17. □

Even if all previous special cases used $\mathcal{X} = \mathbb{R}^n$, the general framework in Chapter 2 also covers the case $\mathcal{X} = \mathbb{C}^n$, see also Remark 2.1. Consequently, by Theorem 2.10 uniform recovery of sparse vectors $x \in \mathbb{C}^n$ with constant modulus constraints using the minimization problem

$$\min\{\|z\|_1 \,:\, Az = Ax,\ |z_j| \in \{0,1\},\, j \in [n],\ z \in \mathbb{C}^n\} \tag{3.25}$$

is characterized by (NSP$^{\mathcal{C}}$) for the set $\mathcal{C} = \{x \in \mathbb{C}^n \,:\, |x_j| \in \{0,1\},\, j \in [n]\}$. Analogously to the previous sections, the NSP condition can be simplified as shown in the following theorem.

**Theorem 3.25.** *Let $\mathcal{C} = \{x \in \mathbb{C}^n \,:\, |x_j| \in \{0,1\},\, j \in [n]\}$, $s \geq 0$ and $A \in \mathbb{C}^{m\times n}$. Then, every $s$-sparse $x \in \mathcal{C}$ is the unique optimal solution of* (3.25) *if and only if*

$$-v_{\overline{S}} \in \mathcal{C} \implies \sum_{j\in S} \chi_{\{|v_j|=1\}} < \sum_{j\in\overline{S}} \chi_{\{|v_j|=1\}} \tag{3.26}$$

*holds for all $v \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ with $v \neq 0$ and all $S \subseteq [n]$, $|S| \leq s$. Here, $\chi_{\{|v_j|=1\}}$ denotes the indicator function of the event $\{|v_j| = 1\}$, that is, $\chi_{\{|v_j|=1\}} = 1$ if $|v_j| = 1$ and $\chi_{\{|v_j|=1\}} = 0$ else. Note that $\sum_{j=1}^n \chi_{\{|v_j|=1\}} = \|v\|_1$, since $v \in \mathcal{C}$.*

*Proof.* We need to show that (NSP$^{\mathcal{C}}$) for $\mathcal{C} = \{x \in \mathbb{C}^n \,:\, |x_j| \in \{0,1\},\, j \in [n]\}$ is equivalent to the NSP (3.26). The statement then directly follows from Theorem 2.10. In order to prove the equivalence between the two NSP conditions, let first $v \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ with $v \neq 0$ and $S \subseteq [n]$ with $|S| \leq s$ and $-v_{\overline{S}} \in \mathcal{C}$. Let $\hat{v}^{(1)}, \hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ and $\|\hat{v}^{(2)}\|_1$ minimal. Since $v \in \mathcal{C} + (-\mathcal{C})$, at least one such decomposition exists. Applying (NSP$^{\mathcal{C}}$) yields

$$\|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 < \|v_{\overline{S}}\|_1. \tag{3.27}$$

Additionally, we define

$$w_j^{(1)} := \begin{cases} 0, & \text{if } v_j = 0,\\ v_j, & \text{if } |v_j| = 1,\\ \hat{v}_j^{(1)}, & \text{otherwise,}\end{cases} \qquad w_j^{(2)} := \begin{cases} 0, & \text{if } v_j = 0,\\ 0, & \text{if } |v_j| = 1,\\ \hat{v}_j^{(2)}, & \text{otherwise,}\end{cases}$$

so that clearly $w^{(1)}, w^{(2)} \in \mathcal{C}$ and $v = w^{(1)} - w^{(2)}$. By Assumption (A4) for $v$, $w^{(1)}, w^{(2)} \in \mathcal{C}$ and the minimal decomposition $\hat{v}^{(1)}, \hat{v}^{(2)} \in \mathcal{C}$, we have

$$0 \leq \|w^{(2)}\|_1 - \|w^{(1)}\|_1 + \|\hat{v}_S^{(1)}\|_1 - \|\hat{v}_S^{(2)}\|_1 - \|v_{\overline{S}}\|_1 + 2\|w_{\overline{S}}^{(1)}\|_1. \tag{3.28}$$

Combining (3.27) and (3.28) shows

$$\begin{aligned}
0 &< \|w^{(2)}\|_1 - \|w^{(1)}\|_1 + 2\|w^{(1)}_{\overline{S}}\|_1 \\
&= \|w^{(2)}_S\|_1 - \|w^{(1)}_S\|_1 + \|w^{(2)}_{\overline{S}}\|_1 + \|w^{(1)}_{\overline{S}}\|_1 \\
&= -\sum_{j \in S \,:\, |v_j| = 1} |v_j| - \sum_{j \in S \,:\, |v_j| \notin \{0,1\}} \big(|\hat{v}^{(1)}_j| - |\hat{v}^{(2)}_j|\big) + \sum_{j \in \overline{S}} |v_j| \\
&= -\sum_{j \in S} \chi_{\{|v_j|=1\}} + \sum_{j \in \overline{S}} \chi_{\{|v_j|=1\}},
\end{aligned}$$

since $-v_{\overline{S}} \in \mathcal{C}$ implies $|v_j| \in \{0,1\}$ for all $j \in \overline{S}$ and since $|v_j| \notin \{0,1\}$ for $j \in S$ implies that $\hat{v}^{(1)}_j$, $\hat{v}^{(2)}_j \neq 0$ and thus $|\hat{v}^{(1)}_j| = |\hat{v}^{(2)}_j| = 1$. This proves (3.26).

For the reverse implication, let $v \in \operatorname{null}(A) \cap (\mathcal{C} + (-\mathcal{C}))$ with $v \neq 0$ and $S \subseteq [n]$ with $|S| \leq s$ and $-v_{\overline{S}} \in \mathcal{C}$. Furthermore, let $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$ with $v = \hat{v}^{(1)} - \hat{v}^{(2)}$ and $\|\hat{v}^{(2)}\|_1$ minimal. Then,

$$\begin{aligned}
\|\hat{v}^{(1)}_S\|_1 - \|\hat{v}^{(2)}_S\|_1 &= \sum_{j \in S} \big(\chi_{\{|\hat{v}^{(1)}_j|=1\}} - \chi_{\{|\hat{v}^{(2)}_j|=1\}}\big) \\
&= \sum_{j \in S \,:\, |\hat{v}^{(1)}_j|=1,\, \hat{v}^{(2)}_j=0} 1 - \sum_{j \in S \,:\, \hat{v}^{(1)}_j=0,\, |\hat{v}^{(2)}_j|=1} 1 \\
&\leq \sum_{j \in S} \chi_{\{|v_j|=1\}} \\
&< \sum_{j \in \overline{S}} \chi_{\{|v_j|=1\}} = \|v_{\overline{S}}\|_1,
\end{aligned}$$

where we used the NSP (3.26) for the last inequality and the assumption $-v_{\overline{S}} \in \mathcal{C}$ for the last equality. The second equality follows from $\hat{v}^{(1)}$, $\hat{v}^{(2)} \in \mathcal{C}$, so that $|\hat{v}^{(1)}_j|$, $|\hat{v}^{(2)}_j| \in \{0,1\}$ for all $j \in [n]$. Thus, the two NSP conditions are equivalent. □

A direct extension to stable and robust recovery as for all previous special cases seems not to be possible, since Assumption (A5) is not satisfied, due to the complicated structure of the minimal decomposition of $v \in \mathcal{C} + (\mathcal{C})$. In contrast to box-constrained integral vectors, even a variable splitting does not lead to a setting where all assumptions are satisfied. Thus, the statements from Section 2.3 concerning stable and robust recovery cannot be applied. Nevertheless, analogously to Theorem 3.25 it can be shown that ($\mathrm{rNSP}^{\mathcal{C}}_{\rho,\tau}$) is equivalent to the condition

$$-v_{\overline{S}} \in \mathcal{C} \implies \sum_{j \in S} \chi_{\{|v_j|=1\}} < \rho \sum_{j \in \overline{S}} \chi_{\{|v_j|=1\}} + \tau |||Av||| \qquad (3.29)$$

for all $v \in (\mathcal{C} + (-\mathcal{C})) \setminus \{0\}$ and all $S \subseteq [n]$ with $|S| \leq s$, where $\rho \in (0,1)$ and $\tau > 0$. The corresponding robust recovery problem becomes

$$\min \{\|z\|_1 \,:\, |||Az - y||| \leq \eta,\ |z_j| \in \{0,1\},\ j \in [n],\ z \in \mathbb{C}^n\} \tag{3.30}$$

for $y \in \mathbb{C}^n$. However, it is not clear, if the NSP (3.29) yields an error bound for recovery of an $s$-sparse $x \in \mathcal{C}$ using (3.30) with $y = Ax + e$ and $|||e||| \leq \eta$.

The side constraint $x \in \mathcal{C}$, i.e., $|x_j| \in \{0,1\}$ for all $j \in [n]$, has a particularly interesting implication. Namely, for $x \in \mathcal{C}$, the ordinary $\ell_1$-norm and the $\ell_0$-norm coincide. Thus, the robust recovery program (3.30) can equivalently be formulated as

$$\min \{\|z\|_0 \,:\, |||Az - y||| \leq \eta,\ |z_j| \in \{0,1\},\ j \in [n],\ z \in \mathbb{C}^n\}. \tag{3.31}$$

By choosing $|||\cdot||| = \|\cdot\|_2$, $\eta = \sqrt{\delta}$ and adapting the notation, we directly obtain the joint antenna selection and phase-only beamforming problem (3.24). Here, $\sqrt{\delta}$ is the given threshold which should be satisfied by the error between the desired and actual output of the users. We now consider the important question of how to solve the resulting recovery problem (3.24), which is equivalent to (3.30) and (3.31). First of all, note that the constant modulus constraint implies that both (3.30) and (3.31) are nonconvex and thus hard to solve in practice. This also shows that in this case, using a convex objective function does not change the hardness of the problem.

## 3.3.2 Solving Problems with Constant Modulus Constraints

Let us come back to the problem of joint antenna selection and phase-only beamforming. Recall that in the network, there are $K$ users to be served and $N$ antenna elements, so that $A \in \mathbb{C}^{K \times N}$, $x \in \mathbb{C}^N$ with $|x_j| \in \{0,1\}$ and $y \in \mathbb{C}^K$. In the following, we describe an algorithmic approach to solve Problem (3.24) to global optimality by exploiting the special structure of the additional constant modulus constraint $|x_j| \in \{0,1\}$.

In order to model Problem (3.24), auxiliary binary variables $b_j$ can be used. This leads to the formulation

$$\begin{aligned} \min_{x \in \mathbb{C}^N} \quad & \sum_{j=1}^{N} b_j \\ \text{s.t.} \quad & \|y - A^\top x\|_2 \leq \sqrt{\delta}, \\ & |x_j| = b_j \qquad \forall j \in [N], \\ & b_j \in \{0,1\} \qquad \forall j \in [N]. \end{aligned} \tag{3.32}$$

In order to obtain a real-valued formulation of Problem (3.32), we introduce the variables $w_j := \mathrm{Re}[x_j]$ and $z_j := \mathrm{Im}[x_j]$ for all $j \in [N]$. Then, let $w = (w_1, \dots, w_N)^\top$ and $z = (z_1, \dots, z_N)^\top$, so that Problem (3.32) can equivalently be written as

$$
\begin{aligned}
\min_{x \in \mathbb{C}^N} \; & \sum_{j=1}^{N} b_j && \text{(3.33a)}\\
\text{s.t.} \; & \sum_{k=1}^{K} \Big(\mathrm{Re}[y_k] - \big(\mathrm{Re}[a_k]^\top w - \mathrm{Im}[a_k]^\top z\big)\Big)^2 \\
& \quad + \Big(\mathrm{Im}[y_k] - \big(\mathrm{Re}[a_k]^\top z + \mathrm{Im}[a_k]^\top w\big)\Big)^2 \le \delta, && \text{(3.33b)}\\
& \mathrm{Re}[x_j]^2 + \mathrm{Im}[x_j]^2 \le b_j \quad \forall\, j \in [N], && \text{(3.33c)}\\
& \mathrm{Re}[x_j]^2 + \mathrm{Im}[x_j]^2 \ge b_j \quad \forall\, j \in [N], && \text{(3.33d)}\\
& b_j \in \{0, 1\} \qquad \forall\, j \in [N], && \text{(3.33e)}
\end{aligned}
$$

where the constant modulus constraints $|x_j|^2 = \mathrm{Re}[x_j]^2 + \mathrm{Im}[x_j]^2 = b_j$, $j \in [N]$ are replaced by the two inequality constraints (3.33c) and (3.33d). Thus, Problem (3.33) is a (nonconvex) mixed-integer nonlinear program (MINLP) with binary variables. The quadratic constraints (3.33b) and (3.33c) can be rewritten as second order cone (SOC) constraints, and thus are convex constraints. The nonconvexity of the problem is due to the quadratic constraints (3.33d). In general, MINLPs can be solved by using spatial branch-and-bound, see, e.g., Vigerske and Gleixner [250]. In this approach, branching is performed on integral and continuous variables, and in each node of the branch-and-bound tree, a continuous relaxation of the problem is solved. This relaxation can be strengthened using gradient cuts for convex constraints and more general linear cuts for other types of constraints. Binary variables with a fractional solution value in the current relaxation lead to the creation of the two new branching nodes, in which the variables are fixed to 0 and 1, respectively. Violated nonlinear constraints are handled by creating branches on continuous variables. In this case, the feasible region is subdivided into two (or possibly more) parts, hence the name spatial branching. Reducing feasible regions then allows to induce strengthened variable bounds, which is called domain propagation. Strengthened bounds in turn lead to tighter relaxation solutions. This spatial branch-and-bound approach is guaranteed to terminate in finite time and to converge to a global optimum under appropriate assumptions, if the concept of $\varepsilon$-$\delta$-feasibility is used, see, e.g., Horst and Tuy [132].

**General Algorithmic Description** In the following, we will exploit the particular structure of Problem (3.33) to enhance the general spatial branch-and-bound proce-

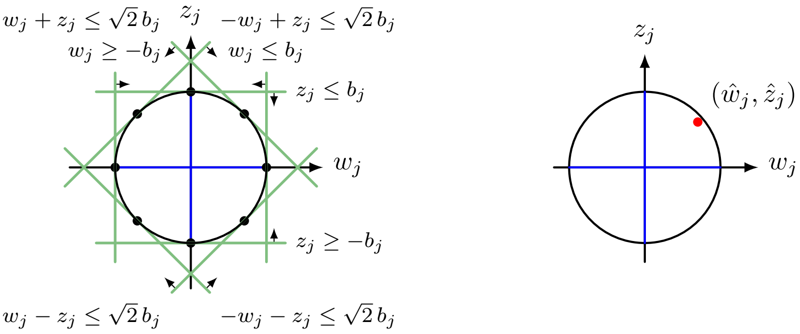

$w_j + z_j \leq \sqrt{2}\, b_j$
$z_j$
$-w_j + z_j \leq \sqrt{2}\, b_j$
$w_j \geq -b_j$
$w_j \leq b_j$
$z_j \leq b_j$
$w_j$
$z_j \geq -b_j$
$w_j - z_j \leq \sqrt{2}\, b_j$
$-w_j - z_j \leq \sqrt{2}\, b_j$
$z_j$
$(\hat{w}_j, \hat{z}_j)$
$w_j$

**Figure 3.2.** Left: Linear inequalities for strengthening the relaxation. Right: Modulus constraint subdivision into orthants.

dure. Namely, we describe a customized domain propagation and branching routine on continuous variables using the constant modulus constraints. Additionally, a simple greedy heuristic to produce feasible solutions for Problem (3.33) is presented. This heuristic can produce upper bounds, which in turn allows to prune nodes in the branch-and-bound tree if the solution value of the relaxation is larger than the current best upper bound.

In each node of the branch-and-bound tree, an LP relaxation of Problem (3.33) is solved, where the binary constraints on $b_j$ are relaxed to $0 \leq b_j \leq 1$, for $j \in [N]$, and the quadratic constraints (3.33b) to (3.33d) are omitted. This LP relaxation is strengthened by adding the following linear inequalities:

$$\begin{aligned} -b_j \leq w_j &\leq b_j, & -b_j \leq z_j &\leq b_j, \\ w_j + z_j &\leq \sqrt{2}\, b_j, & w_j - z_j &\leq \sqrt{2}\, b_j, \\ -w_j + z_j &\leq \sqrt{2}\, b_j, & -w_j - z_j &\leq \sqrt{2}\, b_j, \end{aligned}$$

see Figure 3.2 for a visualization. These linear inequalities approximate the constraints (3.33c). The reason for using an LP relaxation instead of a more general convex relaxation in the nodes of the branch-and-bound tree is that LPs allow for warm-starting using the dual simplex algorithm, see, e.g., Schrijver [218]. Since the quadratic constraints (3.33b) to (3.33d) and the binary constraints on $b_j$ are not present in the LP relaxation, an optimal solution of the LP relaxation violates these constraints in general. For each binary variable with a fractional solution value in the current LP relaxation, this violation is handled by generating two child nodes, as described above. This tightens the LP relaxation in both child nodes.

The error bounding constraint (3.33b) and the constraints (3.33c), which model the upper bound part of the constant modulus constraints are convex SOC constraints, as already mentioned above. Violations of these SOC constraints can be handled by adding a (linear) gradient cut to the LP relaxation, which cuts off the current LP relaxation solution, see Vigerske [249]. Thus it remains to enforce the nonconvex lower bound part of the constant modulus constraints, that is, constraints (3.33d). Due to their nonconvexity, these constraints are harder to enforce than the SOC constraints and the binary constraints. If the solution of the current LP relaxation violates at least one of these constraints, we generate branching nodes, add linear cuts or propagate domains of variables appearing in the violated modulus constraint. This is described in the following in more detail.

## Handling Modulus Constraints

Assume that $(\hat{w}, \hat{z}, \hat{b})$ is the solution of the current LP relaxation of Problem (3.33) and suppose that this solution violates the constraint $w_j^2 + z_j^2 \geq b_j$ for some $j \in [N]$. This violation can then be resolved by one (or more) of the following steps. Note that we do not assume $\hat{b}_j$ to be integral.

1. If the binary variable $\hat{b}_j$ is already fixed to zero, the inequality $w_j^2 + z_j^2 \leq b_j$ implies that we can set $\hat{w}_j, \hat{z}_j$ to zero as well.
2. If the bounds of the continuous variables $w_j$ and $z_j$ are not yet restricted to one of the orthants w.r.t. $w_j \times z_j$, four branching nodes are generated, one for each orthant. That is, the additional constraints $w_j \geq 0$, $z_j \geq 0$ are added to the first node, the constraints $w_j \geq 0$, $z_j \leq 0$ are added to the second node, the constraints $w_j \leq 0$, $z_j \leq 0$ to the third node and the constraints $w_j \leq 0$, $z_j \geq 0$ to the fourth node. Thus, the feasible solution set is subdivided into these four orthants, see Figure 3.2. If either $w_j$ or $z_j$ is already bounded to be nonnegative or nonpositive, then only two of the four orthants can contain feasible solutions. Thus, only two branching nodes are generated in this case.
3. If the bounds of the continuous variables $w_j$ and $z_j$ are already restricted to one of these four orthants, the following domain propagation, separation or branching can be applied. We assume w.l.o.g. that $(\hat{w}_j, \hat{z}_j, \hat{b}_j)$ is feasible for the first orthant, i.e., $w_j \geq 0$ and $z_j \geq 0$.
   (i) **Propagation**: Let $l_1 \leq w_j \leq u_1$, $l_2 \leq z_j \leq u_2$ denote the current lower and upper bounds of the variables $w_j$ and $z_j$, respectively. Define the function $f(x) = \sqrt{1 - x^2}$. Then compute the four points $(l_1, f(l_1))$, $(u_1, f(u_1))$, $(f(l_2), l_2)$ and $(f(u_2), u_2)$ on the unit circle that correspond to the respective lower and upper bounds of $w_j$ and $z_j$. The lower and upper bounds of $w_j$ and $z_j$ can now be strengthened by using these four points. In order for an optimal solution $(w^\star, z^\star, b^\star)$ to fulfill the modulus constraint $w_j^2 + z_j^2 \geq b_j$, the point $(w_j^\star, z_j^\star)$ needs to lie on or above the arc

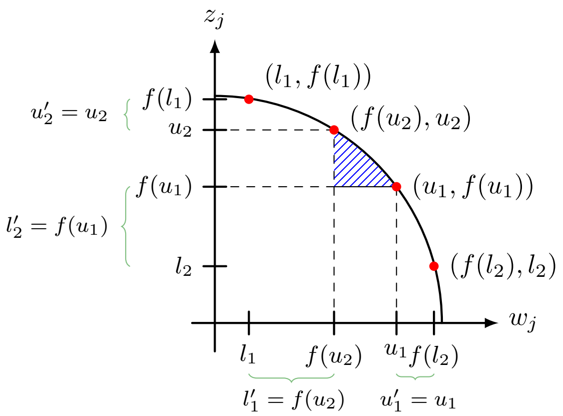


**Figure 3.3.** Bound propagation for the continuous variables appearing in modulus constraints.

between the two points $(l_1', u_2')$ and $(u_1', l_2')$ if $b_j^\star = 1$, where

$$l_1' = \max\{l_1, f(u_2)\}, \quad u_1' = \min\{u_1, f(l_2)\},$$
$$l_2' = \max\{l_2, f(u_1)\}, \quad u_2' = \min\{u_2, f(l_1)\}.$$

Thus, the four values $l_1'$, $u_1'$, $l_2'$ and $u_2'$ can now be used as new and possibly strengthened lower and upper bounds of $w_j$ and $z_j$, respectively. If the binary variable $b_j$ is not yet fixed to one, only the upper bounds are propagated, as $b_j$ could be set to zero in an optimal solution, which would imply $w_j = z_j = 0$ as well. Figure 3.3 visualizes this propagation step.

(ii) **Separation**: If $\hat{w}_j + \hat{z}_j < \hat{b}_j$, the cut $w_j + z_j \geq b_j$ is added to the LP relaxation. Note that it may be reasonable to only add this cut if the violation of the solution of the current LP relaxation is sufficiently large, that is $\hat{w}_j^2 + \hat{z}_j^2 < 1 - \varepsilon$, for some small $\varepsilon > 0$, e.g., $\varepsilon = 10^{-5}$. Otherwise, standard branching rules for handling quadratic constraints can be applied. The cut $w_j + z_j \geq b_j$ is satisfied by every solution on the unit circle in this orthant.

(iii) **Branching**: If $\hat{w}_j + \hat{z}_j \geq \hat{b}_j$, two branching nodes are created, defined by inequalities $f_j\, w_j + g_j\, z_j \geq b_j$. The values $f_j \in \mathbb{R}$ and $g_j \in \mathbb{R}$ can be computed according to Figure 3.4. Note that $w_j + z_j \geq \sqrt{2} b_j$ is an outer approximation of the unit circle which is always valid. The cut $f_j\, w_j + g_j\, z_j \geq b_j$ according to Figure 3.4 is only valid in its corresponding branch.

Within a spatial branch-and-bound framework it makes sense to enforce the nonconvex modulus constraints only when all other constraints are feasible. A modulus constraint $w_j + z_j \geq b_j$ with $j \in [N]$ is selected for enforcing by a "most infeasible"

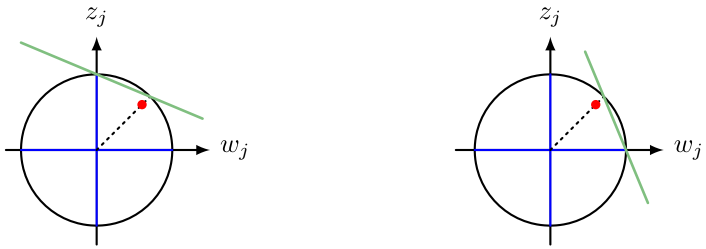


**Figure 3.4.** Inequalities that are added to the sub-nodes.

rule. Therefore, the following measure for the violation of a modulus constraint can be used:

$$\rho(j) = \hat{b}_j - (\hat{w}_j^2 + \hat{z}_j^2),$$

i.e., modulus constraint with index $\bar{j} \in [N]$ so that $\rho(\bar{j})$ is maximal is chosen to be enforced. Algorithm 1 summarizes the whole solving procedure. This procedure is complete in the following sense: The algorithm will terminate with a point $(\hat{w}, \hat{z}, \hat{b})$ such that $\hat{w}_j = \hat{z}_j = 0 = \hat{b}_j$ or $1 - \varepsilon \leq \hat{w}_j^2 + \hat{z}_j^2 \leq 1$ and $\hat{b}_j = 1$. The next subsection presents a simple greedy heuristic to produce feasible solutions.

**Algorithm 1:** Node solving procedure within the branch-and-bound tree

```
Input: Node of the branch-and-bound tree with current LP relaxation of the
       problem including all previously generated cuts, propagated domains
       and previously computed bounds on the objective value
obtain solution $(\hat{w}, \hat{z}, \hat{b})$ of LP relaxation;
if $\hat{b}$ is not integral then
    branch on a fractional binary variable and continue with another node;
else if root-mean error constraint (3.33b) is violated or $\hat{w}_j^2 + \hat{z}_j^2 > \hat{b}_j$ for
  some $j$ then
    call quadratic constraint handler and possibly continue with another
      node;
else if $\hat{w}_j^2 + \hat{z}_j^2 < \hat{b}_j$ then
    call modulus constraint handler to propagate bounds or branch according
      to Section Handling Modulus Constraints and continue with another
      node;
else
    $(\hat{w}, \hat{z}, \hat{b})$ is optimal for the current node;
end
```

## Heuristic Method for Constant Modulus Problems

In this subsection, we present a low-complexity suboptimal heuristic for Problem (3.33), which is inspired by Shechtman et al. [222] and Studer et al. [232].

The heuristic starts with $M = 1$ nonzero element in the vector $x \in \mathbb{C}^n$ and greedily increases the number of nonzeros in $x$ by one in each iteration (in an outer loop) until the root mean-square error bound constraint (3.33b) is met. It is also possible to start with $M = M^{\text{guess}} > 1$ when a reasonable guess is possible or if we have any a priori knowledge about the sparsity of $x$. For each value of $M$, a large number (max_Iter) of suboptimal solutions $\gamma_j$ is computed, by repeatedly initializing $x$ at random. These solutions $\gamma_j$ can be computed in parallel to speed up the heuristic. The vector $x$ is then updated in every iteration (in an inner loop) such that the root-mean square error $e$ between the desired vector $y$ and the current iterate $Ax$ is decreasing. If the smallest error $e_{j^\star}$ among all obtained errors in the max_Iter iterations is smaller than $\sqrt{\delta}$, the corresponding solution $\gamma_{j^\star}$ is reported as the solution of the heuristic. The resulting algorithm is illustrated in Algorithm 2.

## 3.3.3 Numerical Experiments

In this section, we evaluate the performance of the proposed modulus handling based optimal method and suboptimal heuristic method for the problem of joint antenna selection and phase-only beamforming considered in [97], which we described in the beginning of Section 3.3. Recall that the goal is to minimize the number of required phase shifters, to find an assignment of antenna elements to the phase shifters and to design the optimal phase values for the transmit signal vector. In doing so, a given error bound for the output at the users needs to be fulfilled. This leads to Problem (3.24). We refer to [97] for the exact setup of the computations, and only list the used values $N \in \{16, 32, 48, 64\}$ for the number of antennas, $K \in \{2, 3, 4\}$ for the number of users and $\delta^2 \in \{0.1q, 0.2q\}$ for the error bounds, where $q = 1.414$. For each combination of $(N, K, \delta^2)$, we create two instances. We implemented the algorithmic approach described throughout Section 3.3.2 in C using SCIP 4.0.1 [169] and CPLEX 12.7.1 [133] as LP solver. With this setup, we solved Problem (3.33) on a Linux cluster with 3.5 GHz Intel Xeon E5-1620 Quad-Core CPUs, having 32 GB main memory and 10 MB cache. All computations were performed single-threaded with a time limit of one hour (3600 s). The results are shown in Table 3.2.

The table shows the number of branch-and-bound nodes (#n) and the solving time in seconds (t(s)) for four different algorithm variants. The first column block lists the values used for the instance, where 1 and 2 denote the first and the second instance, respectively, for the used combination of values. The second column block displays the results of the default version of SCIP, which applies no special

**Algorithm 2:** Suboptimal heuristic

```
Input: $A$, $y$, $\delta$
Initialize $M \leftarrow 1$ (or $M^{\text{guess}}$);
repeat
    for $i = 1$ to $max\_Iter$ do
        randomly initialize $x \in \mathbb{C}^n$ such that $\|x\|_0 = M$ and $|x_j| \in \{0, 1\}$,
         $j \in [n]$;
        compute error $e = \|y - A^\top x\|_2$;
        for $count = 1$ to $max\_Count$ do
            assign $z \leftarrow x$;
            randomly select two integers $u$ and $v$ such that $u$, $v \in [n]$, $|z_u| = 1$
             and $z_v = 0$;
            compute the residual $r = y - A^\top z + z_u(a^u)^\top$;
            $[z_u^\star, z_v^\star] = \underset{\bar{z}_u, \bar{t}_v}{\operatorname{argmin}} \|r - \bar{z}_u(a^u)^\top + \bar{z}_v(a^v)^\top\|_2$;
            if $|z_u^\star| \geq |z_v^\star|$ then
                $z_u \leftarrow \frac{z_u^\star}{|z_u^\star|}$;
            else
                $z_u \leftarrow 0$ and $z_v \leftarrow \frac{z_v^\star}{|z_v^\star|}$;
            end
            compute $\hat{e} = \|y - A^\top z\|_2$;
            if $\hat{e} < e$ then
                update $x \leftarrow z$ and $e \leftarrow \hat{e}$;
            end
        end
        $\gamma_i \leftarrow x$ and error $E_i \leftarrow e$;
    end
    compute $i^\star = \operatorname{argmin}_i E_i$;
    $E^\star \leftarrow E_{i^\star}$ and $\tilde{x} \leftarrow \gamma_{i^\star}$;
    $M \leftarrow M + 1$;
until $E^\star \leq \sqrt{\delta}$ or $M > N$;
return $\tilde{x}$
```

methods to handle modulus constraints, i.e., they are handled like general quadratic constraints. In the third block, we present the results when the methods for handling modulus constraints as described in Section 3.3.2 are included in SCIP as a constraint handler. In the fourth and fifth block, the results of the same two methods as before are shown, but an initial (not necessarily optimal) solution is computed with the suboptimal greedy heuristic method presented in Section 3.3.2

and passed to the exact solution method. For the number of iterations of the two inner loops, we chose max_Iter = max_Count = 1000. In all four runs, the reading times of the problem files are included in the solving times, as are the runtimes of the suboptimal heuristic in the third and fourth column block. The last column block shows the sparsity $M$ of the solution $\tilde{x}$ computed by the suboptimal heuristic compared to the optimal solution $x^*$ computed by SCIP, as well as the solving time of the suboptimal heuristic.

The end of the table presents the geometric means (GM), shifted geometric means (Shifted GM) and arithmetic means (AM) of the number of nodes and the solving time, as defined in Section 1.3, see (1.2). It turns out that the default version of SCIP already performs quite well. For $K = 2$ users, the running times are very fast even for large values of $N$. For $K \in \{3, 4\}$ users and a very small error bound $\delta^2 = 0.1q$, the instances are much harder to solve. From the shifted geometric means presented in the bottom line, it can be seen that adding the modulus constraint handler to SCIP results in a significantly faster running time (about 26 % faster). However, the number of processed nodes does not significantly change. The shifted geometric mean of the number of nodes that were produced by the modulus constraint handler is 787.25, which is about 24 % of the shifted geometric mean of all nodes (3228).

Executing the suboptimal heuristic and passing its solution to SCIP improves the performance on average, even in the default version of SCIP. Most importantly, the number of nodes is reduced significantly, since many nodes of the branch-and-bound tree can be pruned. Note, however, that for the easier problems the suboptimal heuristic consumes almost all of the solving time. Again, adding the modulus constraint handler to SCIP speeds up the solving process (about 15 % and 39 % speed-up compared to the default with and without initial solution, respectively), but the number of nodes does not decrease. The shifted geometric mean of the number of nodes produced by the modulus constraint handler is 255.43. Comparing to the shifted geometric mean of the number of nodes (500) shows that about half of the branching nodes are used to branch on binary variables.

It is worth mentioning that the suboptimal heuristic actually returns the optimal sparsity level in all but four instances. We observe that only for large instances the heuristic is indeed suboptimal, but these instances cannot be solved by SCIP within the time limit, regardless of the handling of the modulus constraints. Interestingly, one of the instances of Table 3.2 for which the heuristic computes a suboptimal solution runs into the time limit with the default version of SCIP when this solution is passed as starting solution. However, if the suboptimal solution is not computed beforehand, SCIP solves this instance in roughly 700 s.

**Table 3.2.** Analysis and performance evaluation of different solution approaches for solving problems with constant modulus constraints.

| instance | | | | default SCIP | | mod handling | | default SCIP + subopt heur | | mod handling + subopt heur | | subopt heur | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | $K$ | $\delta^2$ | # | #nodes | time | #nodes | time | #nodes | time | #nodes | time | $x^*$ | $\tilde{x}$ | time |
| 16 | 2 | $0.1q$ | 1 | 469 | 1.6 | 501 | 1.6 | 7 | 3.5 | 9 | 3.5 | 2 | 2 | 3.29 |
| 16 | 2 | $0.1q$ | 2 | 863 | 2.1 | 418 | 1.1 | 6 | 3.5 | 6 | 3.4 | 2 | 2 | 3.28 |
| 16 | 2 | $0.2q$ | 1 | 11 | 0.5 | 11 | 0.5 | 1 | 3.4 | 1 | 3.3 | 2 | 2 | 3.28 |
| 16 | 2 | $0.2q$ | 2 | 190 | 0.8 | 136 | 0.6 | 1 | 3.3 | 1 | 3.3 | 2 | 2 | 3.29 |
| 16 | 3 | $0.1q$ | 1 | 3 908 | 9.6 | 2 044 | 4.8 | 998 | 14.0 | 1 118 | 11.4 | 4 | 4 | 9.47 |
| 16 | 3 | $0.1q$ | 2 | 1 690 | 4.9 | 2 337 | 4.6 | 996 | 14.0 | 928 | 12.1 | 4 | 4 | 9.42 |
| 16 | 3 | $0.2q$ | 1 | 1 704 | 7.9 | 1 529 | 3.5 | 47 | 8.8 | 71 | 8.7 | 3 | 3 | 7.08 |
| 16 | 3 | $0.2q$ | 2 | 2 688 | 7.5 | 1 720 | 3.3 | 87 | 8.4 | 95 | 8.3 | 3 | 3 | 7.10 |
| 16 | 4 | $0.1q$ | 1 | 25 507 | 121.8 | 22 684 | 40.7 | 2 374 | 27.7 | 2 470 | 20.9 | 5 | 5 | 16.39 |
| 16 | 4 | $0.1q$ | 2 | 2 853 | 10.4 | 2 603 | 6.5 | 95 | 14.9 | 133 | 14.9 | 4 | 4 | 13.04 |
| 16 | 4 | $0.2q$ | 1 | 2 533 | 11.0 | 1 481 | 3.9 | 289 | 15.6 | 393 | 15.6 | 4 | 4 | 13.06 |
| 16 | 4 | $0.2q$ | 2 | 12 676 | 57.2 | 11 591 | 20.7 | 13 095 | 73.6 | 9 637 | 28.2 | 5 | 5 | 16.41 |
| 32 | 2 | $0.1q$ | 1 | 170 | 7.5 | 204 | 9.6 | 4 | 4.2 | 4 | 4.2 | 2 | 2 | 4.04 |
| 32 | 2 | $0.1q$ | 2 | 1 937 | 6.2 | 714 | 2.2 | 2 | 4.2 | 2 | 4.2 | 2 | 2 | 3.97 |
| 32 | 2 | $0.2q$ | 1 | 201 | 2.7 | 203 | 1.4 | 6 | 4.4 | 8 | 4.3 | 2 | 2 | 3.98 |
| 32 | 2 | $0.2q$ | 2 | 101 | 2.8 | 41 | 1.6 | 17 | 4.5 | 11 | 4.4 | 2 | 2 | 3.99 |
| 32 | 3 | $0.1q$ | 1 | 12 910 | 52.1 | 5 582 | 16.5 | 77 | 13.5 | 134 | 12.9 | 3 | 3 | 8.53 |
| 32 | 3 | $0.1q$ | 2 | 313 | 1.7 | 356 | 1.6 | 177 | 13.1 | 210 | 14.3 | 3 | 3 | 8.58 |
| 32 | 3 | $0.2q$ | 1 | 3 387 | 12.0 | 2 671 | 9.6 | 159 | 13.9 | 198 | 13.8 | 3 | 3 | 8.52 |
| 32 | 3 | $0.2q$ | 2 | 1 380 | 7.2 | 2 514 | 10.7 | 143 | 14.1 | 151 | 14.2 | 3 | 3 | 8.61 |
| 32 | 4 | $0.1q$ | 1 | 96 569 | 559.3 | 13 703 | 51.0 | 154 122 | 652.4 | 57 158 | 156.6 | 4 | 5 | 19.59 |
| 32 | 4 | $0.1q$ | 2 | 141 531 | 816.5 | 97 762 | 301.4 | 147 573 | 648.7 | 66 224 | 172.8 | 5 | 5 | 18.29 |
| 32 | 4 | $0.2q$ | 1 | 8 070 | 31.9 | 9 874 | 37.2 | 43 | 15.6 | 42 | 15.0 | 3 | 3 | 11.75 |
| 32 | 4 | $0.2q$ | 2 | 4 879 | 20.5 | 14 224 | 52.9 | 97 | 16.5 | 123 | 16.7 | 3 | 3 | 11.73 |
| 48 | 2 | $0.1q$ | 1 | 496 | 8.7 | 315 | 8.2 | 1 | 4.9 | 1 | 4.9 | 2 | 2 | 4.64 |
| 48 | 2 | $0.1q$ | 2 | 190 | 2.0 | 734 | 5.0 | 9 | 5.1 | 9 | 5.1 | 2 | 2 | 4.59 |
| 48 | 2 | $0.2q$ | 1 | 15 | 1.6 | 447 | 3.1 | 7 | 5.0 | 7 | 5.0 | 2 | 2 | 4.58 |
| 48 | 2 | $0.2q$ | 2 | 201 | 2.9 | 487 | 4.5 | 7 | 4.9 | 7 | 4.9 | 2 | 2 | 4.58 |
| 48 | 3 | $0.1q$ | 1 | 2 256 | 18.9 | 8 253 | 49.2 | 259 | 16.3 | 283 | 16.4 | 3 | 3 | 9.85 |
| 48 | 3 | $0.1q$ | 2 | 39 191 | 236.6 | 5 542 | 31.4 | 285 | 19.7 | 332 | 20.0 | 3 | 3 | 9.90 |
| 48 | 3 | $0.2q$ | 1 | 1 810 | 21.8 | 2 777 | 19.5 | 381 | 20.5 | 484 | 20.8 | 3 | 3 | 9.87 |
| 48 | 3 | $0.2q$ | 2 | 3 648 | 27.2 | 2 207 | 11.2 | 347 | 23.1 | 362 | 22.9 | 3 | 3 | 9.91 |
| 48 | 4 | $0.1q$ | 1 | 104 244 | 992.9 | 56 336 | 354.9 | 168 481 | 2477.0 | 107 320 | 439.4 | 4 | 5 | 22.63 |
| 48 | 4 | $0.1q$ | 2 | 70 950 | 465.3 | 28 474 | 177.2 | 4 895 | 64.2 | 6 357 | 62.8 | 4 | 4 | 17.99 |
| 48 | 4 | $0.2q$ | 1 | 9 764 | 65.4 | 29 631 | 170.3 | 83 | 24.2 | 91 | 24.6 | 3 | 3 | 13.53 |
| 48 | 4 | $0.2q$ | 2 | 56 580 | 373.8 | 67 515 | 348.5 | 9 933 | 86.9 | 11 707 | 76.0 | 4 | 4 | 18.09 |
| 64 | 2 | $0.1q$ | 1 | 397 | 4.2 | 273 | 3.3 | 10 | 5.7 | 10 | 5.7 | 2 | 2 | 5.18 |
| 64 | 2 | $0.1q$ | 2 | 360 | 4.0 | 360 | 3.9 | 11 | 5.7 | 11 | 5.7 | 2 | 2 | 5.17 |
| 64 | 2 | $0.2q$ | 1 | 505 | 4.5 | 931 | 7.0 | 8 | 5.7 | 8 | 5.7 | 2 | 2 | 5.19 |
| 64 | 2 | $0.2q$ | 2 | 476 | 4.8 | 481 | 4.8 | 11 | 6.0 | 11 | 6.0 | 2 | 2 | 5.19 |

continued on next page

| instance | | | | default SCIP | | mod handling | | default SCIP + subopt heur | | mod handling + subopt heur | | subopt heur | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | $K$ | $\delta^2$ | # | #nodes | time | #nodes | time | #nodes | time | #nodes | time | $x^*$ | $\tilde{x}$ | time |
| 64 | 3 | $0.1q$ | 1 | 11 498 | 105.5 | 5 982 | 36.5 | 529 | 33.0 | 568 | 32.6 | 3 | 3 | 11.17 |
| 64 | 3 | $0.1q$ | 2 | 6 464 | 70.6 | 20 879 | 105.7 | 279 | 21.9 | 333 | 21.9 | 3 | 3 | 11.17 |
| 64 | 3 | $0.2q$ | 1 | 830 | 12.2 | 884 | 11.9 | 519 | 34.2 | 512 | 34.5 | 3 | 3 | 11.24 |
| 64 | 3 | $0.2q$ | 2 | 6 626 | 59.2 | 9 821 | 57.0 | 815 | 34.8 | 966 | 34.2 | 3 | 3 | 11.22 |
| 64 | 4 | $0.1q$ | 1 | 217 176 | 3128.0 | 360 779 | 2932.9 | 20 299 | 172.6 | 24 143 | 152.2 | 4 | 4 | 20.38 |
| 64 | 4 | $0.1q$ | 2 | 75 975 | 670.0 | 329 245 | 2982.6 | >157 627 | >3600.0 | 510 570 | 3492.6 | 4 | 5 | 25.50 |
| 64 | 4 | $0.2q$ | 1 | 51 999 | 688.0 | 47 384 | 396.5 | 10 226 | 126.6 | 28 560 | 185.1 | 3 | 4 | 20.58 |
| 64 | 4 | $0.2q$ | 2 | 82 051 | 748.6 | 26 076 | 173.8 | 119 | 35.1 | 123 | 35.4 | 3 | 3 | 15.24 |
| GM | | | | 2 796 | 21.6 | 2 816 | 16.0 | 164 | 19.6 | 178 | 17.3 | | | |
| Shifted GM | | | | 3 308 | 36.5 | 3 228 | 27.5 | 470 | 26.4 | 500 | 22.4 | | | |
| AM | | | | 22 296 | 197.3 | 25 014 | 176.8 | 14 490 | 175.6 | 17 331 | 110.0 | | | |

## 3.4 Concluding Remarks and Outlook

Throughout this chapter, we have seen various explicit settings with interesting side constraints which all fit into the general framework from Chapter 2. This enabled us to derive NSPs for exact uniform and individual recovery as well as NSPs and corresponding error bounds for stable and robust recovery in a unified manner. For constant modulus constraints, we also described a specialized algorithmic approach to solve the resulting recovery problems to optimality. However, it remains open to find an NSP and a corresponding error bound for stable and robust recovery in the constant modulus setting. Of course, there are many more interesting settings and side constraints or sparsity structures which have (or have not yet) been considered in the literature.

For instance, it is possible to extend the block-sparsity structure from Section 3.1.2 by not only requiring that few blocks contain nonzero elements, but also that each nonzero block itself has few nonzeros. This corresponds to a so-called two-level “hierarchical”-sparsity structure as considered by Simon et al. [224] and Sprechmann et al. [225]. This structure can further be generalized by introducing a recursive tree-like structure in which each block is hierarchical sparse itself. As discussed in Section 3.1.1, this concept can also be transferred to block-diagonal matrices by letting all blocks along the diagonal be low-rank matrices. A recursive application of this structure generalizes hierarchical sparsity to block-diagonal matrices. For an overview over hierarchical sparsity and applications in communication scenarios and quantum state tomography, i.e., the recovery of unknown quantum states, see the recent preprint by Eisert et al. [83] and the references therein. Moreover, this sparsity structure also generalizes the concept of “level sparsity”, which has been considered by, e.g., Adcock et al. [5], Bastounis and Hansen [17], and Li and Adcock [156].

Note that Bastounis and Hansen [17] present an explicit NSP for level sparsity. Apart from this hierarchical structure, it also possible to consider other simultaneous structures, such as low-rank matrices which are also sparse. For an overview, see Kliesch et al. [144] or Oymak et al. [196].

Moreover, there also exist different general sparsity structures which generalize block-sparsity. For instance, Baraniuk et al. [16] introduce the concept of general model sparsity, where a vector is presumed to lie in a union of predefined low-dimensional subspaces. Clearly, every $s$-space vector $x \in \mathbb{R}^n$ lies in the union of all $s$-dimensional subspaces of $\mathbb{R}^n$. By disregarding some of these subspaces, additional structure within $x$ can be encoded. This concept of model sparsity is closely connected to the general union of subspaces concept, as treated by, e.g., Blumensath and Davies [28] Eldar and Mishali [89], or Lu and Do [163]. Both the authors in [16] and [89] introduce an adjusted restricted isometry property as a sufficient recovery condition and consider recovery from random (subgaussian) matrices for the concept of model sparsity and union of subspaces, respectively. In [16], the greedy algorithm CoSaMP and iterative hard thresholding are adapted and used for recovery, whereas in [89], an $\ell_{2,1}$-minimization problem similar to (3.8) is employed. Moreover, it is shown that the union of subspaces model is in fact equivalent to block-sparsity under a small assumption, so that our results from Section 3.1.2 apply as well.

It is reasonable to believe that these sparsity structures also fit into the general framework presented in Chapter 2, possibly with (minor) modifications. In order to express that a vector (or matrix) is sparse in more than one sense or in more than one level, multiple projections need to be introduced.

Until now, we have considered various different settings and have presented null space properties for each setting which guarantees successful (uniform or individual) recovery. These results are very appealing from a theoretical point of view, since they give a complete answer to the question of which properties a measurement matrix needs to be satisfied in order to be "useful" in a sense that recovery is possible. However, from a practical point of view, there remain at least two questions, which have not yet been considered. First, can there be (families of) matrices which actually satisfy any NSPs, or are the presented conditions only of theoretical nature? This question is investigated in the next chapter. Besides, for a given fixed measurement matrix an important aspect is to test whether the matrix satisfies an NSP, which is treated in Section 5.1.

CHAPTER 4

# Recovery Under Random Measurements

Throughout the last chapter, we have analyzed recovery conditions for various special cases, which all emerged from our general framework presented in Chapter 2. Along the way, different examples demonstrated that these conditions are not purely theoretical, but that there exist combinations of matrices and sparsity levels which satisfy different versions of the null space properties. This directly leads to the question whether these examples were mere toy examples or if there are many more matrices which satisfy the various conditions for exact, stable and robust uniform as well as individual recovery. For the classical cases of sparse (nonnegative) vectors and low-rank (positive semidefinite) matrices, which served as running example in Chapter 2, this question has been thoroughly investigated in the literature. It turns out that it is extremely difficult to find deterministic matrices that satisfy the classical NSP or other recovery conditions for sparse vectors with number of measurements and sparsity levels close to the theoretically best possible values. For constructions of deterministic matrices satisfying the restricted isometry property, see Bandeira et al. [13] and DeVore [62]. The book by Vidyasagar [248] gives an overview over the deterministic construction of measurement matrices with favorable properties. Interestingly, it can be shown that various types of random matrices allow for exact, stable and robust uniform as well as individual recovery with high probability if the number of rows is large enough, in dependence on the number of columns and the desired sparsity level. Additionally, it is known that for random matrices, the values for the number of rows and the sparsity levels needed for successful recovery are very close to the theoretically best possible values. This result is especially relevant in practical applications of Compressed Sensing, since the number of rows of the measurement matrix denotes the number of measurements that are taken. Most of the time, taking many measurements is costly or simply impractical, so that it is desirable to take as few measurements as possible. Moreover, random measurement matrices are relatively simple to operate in practice. Thus, finding ways to further reduce the number of measurements needed for successful recovery is highly relevant. If additional knowledge is available, this directly leads to the question whether exploiting this knowledge in the recovery process leads to favorable properties. In the last chapters, we have seen specific examples, in which a recovery condition is satisfied when exploiting a present side constraint, and violated otherwise. Moreover, for the case of block-sparse vectors, Theorem 3.14 presents a family of matrices that satisfy the NSP if the nonnegativity is taken into account, but violates the corresponding NSP if the nonnegativity is ignored. All these results indicate that exploiting the side constraints indeed lead to weaker recovery conditions, which can be satisfied by more matrices.

In this chapter, we will discuss recovery of sparse nonnegative vectors under random measurements. We derive a bound for the minimal number of measurements needed for uniform recovery of sparse nonnegative vectors. In the literature, a bound for sparse nonnegative vectors is already available, but it only guarantees uniform recovery asymptotically, i.e., for the dimension $n \to \infty$. In this chapter, we adapt the proof of a well-known bound for sparse vectors to the case of sparse nonnegative vectors, in order to obtain a nonasymptotic bound which guarantees uniform recovery for all dimensions $n$. Unfortunately, it will turn out that the bound is weak, since it is larger than the bound for sparse vectors. This is most likely due to some estimations in the process of obtaining the bounds being too weak. However, we will show empirically and numerically in simulations that the minimal number of measurements needed for uniform recovery for sparse nonnegative vectors is indeed smaller than for sparse vectors. This indicates that our bound is far from being optimal and most likely can significantly be improved. The simulations highlight the effect of nonnegativity as side constraint on the recovery conditions, by showing that more (random) measurement matrices allow for uniform recovery if the nonnegativity is exploited.

Furthermore, we also consider the recovery of block-sparse matrices under random measurements. For block-sparse matrices, we provide the first result that random measurement operators satisfy the corresponding NSP with high probability if the number of measurements is sufficiently large. By that, we show that uniform recovery of block-sparse matrices is possible with high probability under random measurements.

In Section 4.1 we will review the existing literature and introduce the concepts needed for an analysis of recovery under random measurements. The subsequent

Section 4.2 treats the recovery of sparse nonnegative vectors under random measurements and derives a theoretical bound for the number of needed measurements, and compares with empirical bounds obtained by sampling. Section 4.3 then treats the recovery of block-diagonal matrices from Section 3.1 without the positive semidefiniteness constraint. Lastly, Section 4.4 provides an outlook. Throughout this chapter, we again only consider exact uniform and individual recovery. As before, the statements can easily be adapted to also cover stability and robustness.

## 4.1 Recovery Under Random Measurements – An Overview

The first result that random matrices satisfy a recovery guarantee is due to Candès and Tao [38, 45], where the authors show that Gaussian random matrices satisfy the RIP. Shortly after, Mendelson et al. [178] and Baraniuk et al. [15] extended this result to subgaussian random matrices, Bernoulli random matrices and other matrices satisfying a certain concentration inequality. A crucial tool to show that random matrices satisfy an NSP is Gordon's "Escape Through a Mesh", which first appears in Gordon [121], and is used by Rudelson and Vershynin [215] for the first time in Compressed Sensing in order to show that Gaussian random matrices satisfy the NSP with high probability, given that the number of measurements, that is, the number of rows of the measurement matrix is large enough in comparison to the number of columns and the sparsity level. Stojnic [226] uses Gordon's Escape theorem together with duality to obtain tight estimations for when Gaussian random matrices satisfy the NSPs for individual and uniform recovery of sparse vectors as well as for individual recovery of sparse nonnegative vectors. A simpler analysis which yields slightly less precise estimations for the NSP of Gaussian random matrices is contained in Foucart and Rauhut [104], where Gordon's Escape theorem is combined with conic duality. Gordon's Escape theorem has also been successfully applied to individual and uniform recovery in various other settings, e.g., block-sparse vectors [229, 230] and low-rank matrices [139, 193, 195], whereas Kabanava et al. [139] uses the approach of Rudelson and Vershynin. Oymak and Hassibi [193] and Oymak et al. [195] extend the approach of Stojnic. For an overview over low-rank matrix recovery, see Davenport and Romberg [61]. Liaw et al. [158] extend Gordon's Escape theorem also to subgaussian matrices, since in its original form, it can only be applied for Gaussian random matrices. Mendelson's "Small Ball Method" is another technique which can be used for various other types of random matrices, see Dirksen et al. [66] for an overview over the applicability.

The publications [9, 44, 49] provide different frameworks to obtain estimations for individual recovery in various settings, see also Tropp [241] for an overview and an extension of Mendelson's Small Ball Method, the so-called "Bowling Scheme". A prominent application of the results of Chandrasekaran et al. [49] is gridless Compressed Sensing [233], and the results of Amelunxen et al. [9] are applied for binary and, more general, finite valued Compressed Sensing by Keiper et al. [141]. For a general introduction to high-dimensional probability and its use in Compressed Sensing and related areas see Vershynin [246] or Vershynin [247]. The book by Vidyasagar [248] also collects some results on random matrices in Compressed Sensing, and the book [104] contains most of the relevant probabilistic tools needed for the analysis of recovery with Gaussian (and to some extent also subgaussian) random matrices.

In a different direction of work, results on random matrices satisfying the classical and the nonnegative NSP (see (NSP) and (NSP$_{\geq 0}$)) were obtained by Donoho and Tanner in several publications. They reformulated the respective NSP in terms of neighborliness of certain projected polytopes [67, 73, 74], see Proposition 3.12 for the result for the nonnegative NSP. Building on results of Affentranger and Schneider [6], Böröczky and Henk [30] as well as Vershik and Sporyshev [245] about random (projections of) polytopes, they derive bounds for the number of measurements to guarantee uniform and individual recovery [68, 75, 77, 79]. These bounds hold asymptotically for the number of columns tending to infinity and the ratio between the number of rows and the number columns as well as the ratio between the number of rows and the sparsity level remains constant. This asymptotic nature of the results is due to the underlying theory of neighborliness of randomly projected polytopes. Since this direction of work is not treated within this thesis, we refer to Donoho and Tanner [76, 78] and the corresponding paragraph in the notes of Section 9 in [104] for an overview over the obtained results and bounds. It is worth mentioning that the bounds obtained from Donoho and Tanner show that there is indeed a difference between the classical NSP and the nonnegative NSP. For both uniform and individual recovery, the nonnegative NSP is satisfied for a smaller number of random Gaussian measurements compared to the classical NSP.

Overall, for various settings without additional side constraints, such as (block-) sparse vectors and low-rank matrices, precise estimation of the minimal number of measurement needed for a random measurement matrix (or operator) to satisfy the NSPs for individual and uniform corresponding to the respective setting are known in the literature. These results hold for various types of random matrices, such as Gaussian, subgaussian, and also 0/1 Bernoulli matrices. In the presence of additional side constraints, much less is known. For sparse nonnegative vectors, there are asymptotic bounds for individual and uniform recovery available from the polytope analysis of Donoho and Tanner. Oymak and Hassibi [192] derive conditions for individual and uniform recovery of low-rank positive semidefinite matrices based on

Stojnic's involved analysis. Stojnic [226, 229] derives bounds for individual recovery of sparse nonnegative and block-sparse nonnegative vectors, and states that the same analysis can also be applied for uniform recovery of sparse nonnegative vectors, but does not include it into the publication [226], since

> "[...] the analysis of that cases becomes a bit more tedious and certainly loses on elegance."[3]

In the next chapter, we extend the analysis of [104] using conic duality from sparse vectors to sparse nonnegative vectors. Since the nonnegative NSP has a different structure than the classical NSP, namely, invariance under entrywise sign changes cannot be used, the analysis becomes more difficult. Nevertheless, we can derive a bound for the minimal number of measurements (i.e., rows) needed for a Gaussian random matrix to satisfy the nonnegative NSP with high probability. This bound is non-asymptotic in contrast to the estimations that can be obtained from random polytope theory. Unfortunately, the derived bound is larger than the corresponding bound for sparse vectors in [104, Theorem 9.29]. However, numerical experiments explicitly show that recovery of sparse nonnegative vectors needs fewer measurements than recovery of sparse vectors. Moreover, we also provide small simulations which indicate that if the estimations in both bounds are replaced by empirically obtained quantities, then indeed fewer measurements seem to be needed for sparse nonnegative vectors compared to sparse vectors. Hence, the estimations used in this thesis for sparse nonnegative vectors can most likely be improved. Such an improvement remains an interesting open problem.

In order to state the result about random measurements for sparse nonnegative vectors, we first need to introduce some concepts from probability in the following. The definitions and statements are taken from [104]. Note that we assume the reader to be familiar with basics in probability. A good source is the monograph by Ross [213].

Let $X$ be a random variable and let $\mathbb{P}$ be a probability measure on a probability space. The *cumulative distribution function (cdf)* $F = F_X$ of the random variable $X$ is defined as $F(t) := \mathbb{P}(X \leq t)$ for $t \in \mathbb{R}$. If there exists a function $\phi\colon \mathbb{R} \mapsto \mathbb{R}_+$ with

$$\mathbb{P}(a \leq X \leq b) = \int_a^b \phi(t)\, dt$$

for all $a < b \in \mathbb{R}$, then $\phi$ is called the *probability density function (pdf)* of $X$. It then holds

$$\phi(t) = \frac{d}{dt} F(t).$$

[3] Stojnic [226, p. 40]

The *expectation* (or mean) of $X$ is defined as

$$\mathbb{E}[X] := \int_{\Omega} X(\omega)\, d\mathbb{P}(\omega),$$

where $\Omega$ is the sample space of the probability space. Throughout this chapter, we use the Gaussian distribution. If the pdf $\psi$ of a random variable $X$ has the form

$$\psi(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big( - \frac{(t-\mu)^2}{2\sigma^2} \Big),$$

then $X$ is called a *normally distributed* random variable or a *Gaussian* random variable with mean $\mu$ and variance $\sigma^2$, denoted by $X \sim \mathcal{N}(\mu, \sigma^2)$. If $\mu = 0$ and $\sigma^2 = 1$, then $X$ is called a *standard Gaussian* random variable. In the following, we denote with $\varphi$ and $\Phi$ the pdf and cdf of the standard Gaussian distribution, respectively, i.e.,

$$\begin{aligned} \varphi(t) &= \frac{1}{\sqrt{2\pi}} \exp\Big( - \frac{t^2}{2} \Big), \\ \Phi(t) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} \exp\Big( - \frac{x^2}{2} \Big)\, dx. \end{aligned}$$

Let $X_1, \dots, X_n$ be a collection of random variables on a common probability space. If they are independent and all have the same distribution, they are called *independent identically distributed* (i.i.d.). The vector $X = [X_1, \dots, X_n]^\top \in \mathbb{R}^n$ is called a *random vector*. Correspondingly, a *random matrix* $X \in \mathbb{R}^{m \times n}$ is a collection of $mn$ random variables $X_{ij}$ on a common probability space. A *standard Gaussian random vector* is a vector $g \in \mathbb{R}^n$ whose components $g_i$ are i.i.d. standard normal random variables, and a standard Gaussian random matrix $G$ is defined analogously. In the remaining parts of this chapter, unless noted otherwise, $g$ and $G$ will denote a standard Gaussian random vector and random matrix of appropriate dimension, respectively. The expectation of the $\ell_2$-norm of a standard Gaussian random vector will play an important role, so that we denote this quantity by $E_n$. It satisfies the bounds

$$\frac{n}{\sqrt{n+1}} \le E_n = \sqrt{2}\,\frac{\Gamma((n+1)/2)}{\Gamma(n/2)} \le \sqrt{n}, \tag{4.1}$$

see, e.g., Foucart and Rauhut [104, Proposition 8.1]. Additionally, we need the following classical result for concentration of measure. It bounds the probability that a Lipschitz function evaluated at a Gaussian random vector deviates from its expectation.

**Theorem 4.1** ([104, Theorem 8.34], [121, Theorem 3.2])**.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz function with Lipschitz constant $L > 0$, and let $g \in \mathbb{R}^n$ be a standard Gaussian random vector. Then, for all $t > 0$*

$$\mathbb{P}\big(f(g) - \mathbb{E}[f(g)] \geq t\big) \leq \exp\Big(-\frac{t^2}{2L^2}\Big),$$
$$\mathbb{P}\big(|f(g) - \mathbb{E}[f(g)]| \geq t\big) \leq 2\exp\Big(-\frac{t^2}{2L^2}\Big).$$

Next, we define the *Gaussian width* of a set $S \subseteq \mathbb{R}^n$ which plays a crucial role in the derivation of bounds on the number of measurements needed for a random measurement matrix to satisfy an NSP with high probability.

**Definition 4.2.** *The Gaussian width $\omega(S)$ of a set $S \subseteq \mathbb{R}^n$ is defined as*

$$\omega(S) := \mathbb{E}\big[\sup\{g^\top z \,:\, z \in S\}\big],$$

*where $g$ is a standard Gaussian random vector.*

Define the unit sphere $\mathbb{S}^{n-1} := \{x \in \mathbb{R}^n \,:\, \|x\|_2 = 1\}$. Let $T \subseteq \mathbb{S}^{n-1}$ be a closed subset of the unit sphere and $A \in \mathbb{R}^{m\times n}$ be a standard Gaussian random matrix. Then, the function $f\colon A \mapsto \inf\{\|Ax\|_2 \,:\, x \in T\}$ is a Lipschitz function with Lipschitz constant $L = 1$, as the proof of [104, Theorem 9.21] shows. For this function, we obtain the following variant of Theorem 4.1 in terms of the expectation.

**Theorem 4.3** ([49, Theorem 3.2], [121, Corollary 1.2])**.** *Let $T \subseteq \mathbb{S}^{n-1}$ be a closed subset of the unit sphere in $\mathbb{R}^n$, and let $A\colon \mathbb{R}^n \to \mathbb{R}^m$ be a random map with i.i.d. zero-mean Gaussian entries having variance 1. Then,*

$$\mathbb{E}\Big[\inf_{x\in T} \|Ax\|_2\Big] \geq E_m - \omega(T).$$

If $T$ is a cone, the quantity $\inf_{x\in T}\|Ax\|_2$ is also known in the literature as *minimum conic singular value*, see, e.g., Tropp [241]. Theorem 4.1 and the bound for the expectation in Theorem 4.3 leads to the following bound on the probability on a deviation of the function $f$.

**Theorem 4.4** ([104, Theorem 9.21])**.** *Let $A \in \mathbb{R}^{m\times n}$ be a standard Gaussian random matrix, and let $T \subseteq \mathbb{S}^{n-1}$ be a subset of the unit sphere in $\mathbb{R}^n$. Then, for all $t > 0$*

$$\mathbb{P}\Big(\inf_{x\in T} \|Ax\|_2 \leq E_m - \omega(T) - t\Big) \leq \exp\Big(-\tfrac{1}{2}t^2\Big).$$

The result in Theorem 4.4 is known in the literature under the name "Gordon's Escape Through a Mesh". The original form of this statement appears in Gordon [121, Corollary 3.4], and has been refined by Rudelson and Vershynin [215]. Another variant of this result also appears in Stojnic [226]. Gordon's Escape Through a Mesh in Theorem 4.4 is the main ingredient to prove that (Gaussian) random measurement matrices satisfy the NSP with high probability. Recall the definition of the classical null space property (NSP) for characterizing uniform recovery of sparse vectors by $\ell_1$-minimization in Example (2.12.1):

$$\|v_S\|_1 < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\},\ \forall\, S \subseteq [n],\ |S| \le s. \tag{NSP}$$

This condition is clearly invariant under scaling the vectors $v$. Thus, we can assume that $\|v\|_2 = 1$. Consequently, we define the set

$$T_s := \{v \in \mathbb{R}^n \,:\, \|v_S\|_1 \ge \|v_{\overline{S}}\|_1 \text{ for some } S \subseteq [n],\ |S| \le s,\ \|v\|_2 = 1\}. \tag{4.2}$$

A measurement matrix $A \in \mathbb{R}^{m\times n}$ satisfies (NSP) of order $s$, if $\|Ax\|_2 > 0$ for all $x \in T_s$, due to the scaling invariance of the NSP. Thus, if $A$ is a standard Gaussian random measurement matrix and if $E_m - \omega(T_s) - t > 0$, then Theorem 4.4 directly states that the NSP is satisfied with probability at least $1 - \exp(-\frac{1}{2}t^2)$. It now remains to upper bound the Gaussian width $\omega(T_s)$ of the unit-norm vectors violating the NSP condition. In the literature, there exist different approaches to derive bounds for this Gaussian width. Rudelson and Vershynin [215] show that $T_s \subseteq 2\operatorname{conv}\{x \in \mathbb{R}^n \,:\, \|x\|_0 \le s,\ \|x\|_2 = 1\}$, and estimate the Gaussian width of this set. Stojnic [226] uses duality arguments for the underlying linear program in the definition of the Gaussian width to provide sharp bounds for the Gaussian width $\omega(T_s)$. Foucart and Rauhut [104] replace the set $T_s$ by an appropriate cone in order to use conic duality and an outer approximation of the corresponding dual cone to derive a bound on the Gaussian width $\omega(T_s)$.

This general approach using Theorem 4.4 also works for different NSPs, e.g., NSPs that characterize uniform recovery in the presence of different side constraints. In this case, only the set $T_s$ needs to be adapted accordingly, and its Gaussian width needs to be estimated. Of course, this approach also covers robust and stable as well as individual recovery, since in all these cases, again only the set $T_s$ changes. The case of individual recovery can even be made more concrete by noting that the corresponding NSP that characterizes (or sometimes implies) individual recovery is equivalent to the condition that the descent cone (or tangent cone) of the objective function of the corresponding recovery problem at $x^{(0)}$ does not intersect the null space of the measurement matrix $A$, see also Remark 2.35. The probability of this event for a standard Gaussian random measurement matrix $A$ can be bounded by the distance of a standard Gaussian random vector from the dilated subdifferential of the respective objective function at $x^{(0)}$, due to the duality relationship between the tangent cone and the normal cone, which is the cone over the subdifferential. This connection is exploited by Amelunxen et al. [9] and Chandrasekaran et al. [49] in order to derive bounds for the minimal number of measurements needed for guaranteeing individual recovery with high probability in various settings.

In the following, we briefly outline the key points of the analysis of Foucart and Rauhut, since we will closely follow their analysis in our own analysis of sparse nonnegative vectors in Section 4.2. For a vector $v \in \mathbb{R}^n$, we define $v^*$ to be the nonnegative rearrangement of $|v|$, that is $v_1^* \geq \cdots \geq v_n^* \geq 0$ and there exists a permutation $\sigma$ with $v^*_{\sigma(i)} = |v_i|$ for all $i \in [n]$. Furthermore, we define the cones

$$K_s := \Big\{v \in \mathbb{R}^n \,:\, \sum_{i=1}^{s} v_i \geq \sum_{i=s+1}^{n} v_i,\ v_i \geq 0\,\forall\, i \in [n]\Big\},$$
$$Q_s := \{v \in \mathbb{R}^n \,:\, v_1 = \cdots = v_s = t,\ v_i \geq -t,\ i = s+1, \ldots, n \ \text{ for some } t \geq 0\}.$$

Then, [104, Lemma 9.32] shows $Q_s \subseteq K_s^*$, where $K_s^*$ is the dual cone to $K_s$, which is defined as $K_s^* := \{x \,:\, \langle x, z\rangle \geq 0 \,\forall\, z \in K_s\}$. For $K_s$ and its dual cone, we can use the following result about weak conic duality. For a vector $g \in \mathbb{R}^n$ and a cone $K$, we have

$$\max\{\langle g, x\rangle \,:\, \|x\|_2 \leq 1,\ x \in K\} \leq \min\{\|g+z\|_2 \,:\, z \in K^*\}, \tag{4.3}$$

see, e.g., [104, Equation (B.40)]. Since the set $T_s$ as defined in (4.2) is invariant under permutation of the indices and entrywise sign changes, the Gaussian width $\omega(T_s)$ can be written as

$$\omega(T_s) = \mathbb{E}\Big[\max\,\big\{\langle g, z\rangle \,:\, z \in T_s\big\}\Big] = \mathbb{E}\Big[\max\,\big\{\langle g^*, v\rangle \,:\, v \in K_s,\ \|v\|_2 \leq 1\big\}\Big].$$

Using weak conic duality in (4.3) implies

$$\begin{aligned}\omega(T_s) &\leq \mathbb{E}\Big[\min\,\big\{\|g^* + x\| \,:\, x \in K_s^*\big\}\Big]\\ &\leq \mathbb{E}\Big[\min\,\big\{\|g^* + x\| \,:\, x \in Q_s\big\}\Big]\\ &\leq \min_{t\geq 0}\Bigg\{\mathbb{E}\Big[\Big(\sum_{i=1}^{s}\big(g_i^* + t\big)^2\Big)^{1/2}\Big]\\ &\qquad\qquad + \mathbb{E}\Big[\min_{z_{s+1},\ldots,z_n \geq -t}\Big(\sum_{i=s+1}^{n}\big(g_i^* + z_i\big)^2\Big)^{1/2}\Big]\Bigg\}.\end{aligned} \tag{4.4}$$

For a fixed $t \geq 0$, Foucart and Rauhut [104, Section 9.3] obtain the following estimates:

$$E_1^{(\mathrm{lin})} := \mathbb{E}\Big[\Big(\sum_{i=1}^{s} \big(g_i^* + t\big)^2\Big)^{1/2}\Big] \leq t\sqrt{s} + \sqrt{s} + \sqrt{2s \ln\big(\tfrac{en}{s}\big)}, \tag{4.5}$$

$$E_2^{(\mathrm{lin})} := \mathbb{E}\Big[\min_{z_{s+1},\dots,z_n \geq -t} \Big(\sum_{i=s+1}^{n} \big(g_i^* + z_i\big)^2\Big)^{1/2}\Big] \leq \sqrt{\big(n-s\big)\sqrt{\tfrac{2}{\pi e}}\frac{\exp(-t^2/2)}{t^2}}, \tag{4.6}$$

where $e := \exp(1)$. Choosing $t = \sqrt{2\ln(en/s)}$ in (4.5) and (4.6) and inserting the resulting estimates into (4.4) yields the bound

$$\omega(T_s) \leq \sqrt{2s \ln\big(\tfrac{en}{s}\big)}\Big(1 + \frac{1}{\sqrt{2\ln(en/s)}} + \frac{1}{(8\pi e^3)^{1/4}\ln(en/s)}\Big). \tag{4.7}$$

For the details, we refer to [104, Section 9.3]. Gordon's Escape Theorem 4.4 yields the following bound for the minimal number of measurements.

**Theorem 4.5** ([104, Corollary 9.34])**.** *Let $A \in \mathbb{R}^{m\times n}$ be a standard Gaussian random matrix and let $s < n$ as well as $\varepsilon > 0$. If*

$$\frac{m^2}{m+1} \geq \Big(\omega + 2\ln\big(\tfrac{1}{\varepsilon}\big)\Big)^2,$$

*where $\omega$ is the estimation of $\omega(T_s)$ in* (4.7)*, then $A$ satisfies* (NSP) *of order $s$ with probability at least $\varepsilon$.*

## 4.2 Analysis of Random Measurements for Sparse Nonnegative Vectors

In this section, we will derive bounds for the minimal number of measurements $m$ for uniform recovery of sparse nonnegative vectors. The derivation is a direct adaption of the analysis of Foucart and Rauhut [104] outlined above to the setting of sparse nonnegative vectors. Since the nonnegative null space property (NSP$_{\geq 0}$) contains a nonnegativity constraint, the corresponding set $T_s$ of vectors violating this NSP is not invariant under sign changes. Hence, the derivation of bounds for the Gaussian width of $T_s$ is more involved than in the case of sparse vectors. Moreover, it seems that a direct adaption of the appealing geometric approach of embedding the set of vectors violating (NSP$_{\geq 0}$) in an inflated sparse unit-norm ball as done by Rudelson and Vershynin [215] is not straight-forward, since the corresponding ball remains unknown.

Recall that the nonnegative null space property ($\mathrm{NSP}_{\geq 0}$) of order $s$ reads

$$v_{\overline{S}} \leq 0 \implies \sum_{i \in S} v_i < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \mathrm{null}(A)\backslash\{0\},\ \forall\, S \subseteq [n],\ |S| \leq s. \tag{$\mathrm{NSP}_{\geq 0}$}$$

Thus, for any $s \geq 0$, the set $T_s$ of unit-norm vectors violating ($\mathrm{NSP}_{\geq 0}$) is given by

$$T_s \coloneqq \{v \in \mathbb{R}^n \,:\, \|v\|_2 = 1,\ v_{\overline{S}} \leq 0,\ \mathbb{1}^\top v \geq 0 \text{ for some } S \subseteq [n],\ |S| \leq s\}. \tag{4.8}$$

This set $T_s$ is invariant under permutation of the indices, but not invariant under sign changes. Hence, we can order the indices nonincreasingly, but we cannot order the indices nonincreasingly according to their absolute value. For a vector $x \in \mathbb{R}^n$, we call $\tilde{x}$ the nonincreasing rearrangement of $x$, if $\tilde{x}_1 \geq \tilde{x}_2 \geq \cdots \geq \tilde{x}_n$ and there exists a permutation $\tau$ of $[n]$ with $\tilde{x}_i = x_{\tau(i)}$ for all $i \in [n]$.

Note that for the null space property (NSP) for sparse vectors, the set of vectors violating the NSP is invariant under sign changes, so that in this case, the nonincreasing rearrangement of $(|x|_1, \ldots, |x|_n)^\top$ can be used. Since the signs do not play a role, the analysis of the Gaussian width becomes easier in this case.

By using the nonincreasing rearrangement, we can omit the choice of the set $S$ in the definition of the set $T_s$ and arrive at the convex cone $K_s$ defined as

$$K_s \coloneqq \{v \in \mathbb{R}^n \,:\, v_{s+1}, \ldots, v_n \leq 0,\ \mathbb{1}^\top v \geq 0\}.$$

The Gaussian width of $T_s$ can now be computed using the convex cone $K_s$ and the nonincreasing rearrangement $\tilde{g}$ of the standard Gaussian random vector $g \in \mathbb{R}^n$, since we have that

$$\begin{aligned} \omega(T_s) &= \mathbb{E}\big[\max\{\langle g, x\rangle \,:\, x \in T_s\}\big] \\ &= \mathbb{E}\big[\max\{\langle \tilde{g}, x\rangle \,:\, x \in K_s \cap \mathbb{S}^{n-1}\}\big] \\ &\leq \mathbb{E}\big[\min\{\|\tilde{g} + z\|_2 \,:\, z \in K_s^*\}\big], \end{aligned}$$

where $K_s^*$ is the dual cone of $K_s$, and the expectation is taken with respect to $g$. The last inequality follows from weak conic duality in (4.3). The next lemma shows an explicit formulation of the dual cone $K_s^*$.

**Lemma 4.6.** *Let $s \geq 0$ and $K_s = \{v \in \mathbb{R}^n \,:\, v_{s+1}, \ldots, v_n \leq 0,\ \mathbb{1}^\top v \geq 0\}$. Then, the dual cone $K_s^*$ of the convex cone $K_s$ is given by*

$$K_s^* = \{v \in \mathbb{R}^n \,:\, v_i = t\ \forall\, i \in [s],\ v_i \leq t\ \forall\, i \in \{s+1, \ldots, n\} \textit{ for some } t > 0\}.$$

*Proof.* Let

$$Q_s := \{v \in \mathbb{R}^n \,:\, v_i = t \;\forall\, i \in [s],\; v_i \le t \;\forall\, i \in \{s+1,\dots,n\} \text{ for some } t > 0\}.$$

In order to prove the inclusion $Q_s \subseteq K_s^*$, let $z \in Q_s$ and $u \in K_s$. Then,

$$\langle z, u\rangle = \sum_{i=1}^{s} t\, u_i + \sum_{i=s+1}^{n} z_i\, u_i \ge t \cdot \mathbb{1}^\top u \ge 0,$$

so that $z \in K_s^*$ by definition of the dual cone.

For the reverse inclusion, let $z \in K_s^*$. Then, $\langle z, u\rangle \ge 0$ has to hold for all $u \in K_s$. Assume that there exist $i \neq j \le s$ with $z_i \neq z_j$, and suppose $z_i > z_j$ without loss of generality. Then, $w \in \mathbb{R}^n$ with $w_i = -1$, $w_j = 1$ and $w_k = 0$ for all $k \notin \{i, j\}$ is contained in $K_s$, but we have $\langle z, w\rangle = -z_i + z_j < 0$. Thus, $z_i = z_j$ for all $i, j \le s$. Moreover, $z_i \ge 0$ for all $i \le s$, since otherwise the vector $w = (1, 0, \dots, 0)^\top \in K_s$ yields $\langle z, w\rangle < 0$. This shows $z_i = t$ for all $i \le s$ and some $t \ge 0$. For the remaining indices, assume there exists $j \ge s+1$ with $z_j > z_1 = t$. In this case, the vector $w \in \mathbb{R}^n$ with $w_1 = 1$, $w_j = -1$ and $w_k = 0$ for all $k \notin \{1, j\}$ is contained in $K_s$ but again, $\langle z, w\rangle = z_1 - z_j < 0$, a contradiction. Thus, $z_i \le t$ for all $i \ge s+1$, which shows $z \in Q_s$. □

This representation of the dual cone $K_s^*$ implies that we can estimate the Gaussian width $\omega(T_s)$ as follows:

$$\begin{aligned}
&\mathbb{E}\big[\min\{\|\tilde{g} + z\|_2 \,:\, z \in K_s^*\}\big]\\
=&\,\mathbb{E}\Big[\min\Big\{\Big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2 + \sum_{i=s+1}^{n}(\tilde{g}_i + z_i)^2\Big)^{1/2} \,:\, t > 0,\; z_{s+1}, \dots, z_n \le t\Big\}\Big]\\
\le&\,\mathbb{E}\Big[\min\Big\{\Big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2\Big)^{1/2} + \Big(\sum_{i=s+1}^{n}(\tilde{g}_i + z_i)^2\Big)^{1/2} \,:\, t > 0,\; z_{s+1}, \dots, z_n \le t\Big\}\Big].
\end{aligned}$$

Consider a fixed $t > 0$, and define

$$\begin{aligned}
E_1^{(\mathrm{nng})} &:= \mathbb{E}\Big[\Big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2\Big)^{1/2}\Big],\\
E_2^{(\mathrm{nng})} &:= \mathbb{E}\Big[\min_{z_{s+1},\dots,z_n \le t}\Big(\sum_{i=s+1}^{n}(\tilde{g}_i + z_i)^2\Big)^{1/2}\Big].
\end{aligned} \tag{4.9}$$

These terms resemble the terms $E_1^{(\mathrm{lin})}$ and $E_2^{(\mathrm{lin})}$ in (4.5) and (4.6), respectively, which appeared in the analysis for sparse vectors. Besides the slightly different dual cone $K_s^*$, the nonincreasing rearrangement of $g$ is needed for sparse nonnegative vectors, whereas for sparse vectors, the nonincreasing rearrangement of $|g|$ can be used. This implies that the deriving good bounds for $E_1^{(\mathrm{nng})}$ and $E_2^{(\mathrm{nng})}$ becomes more tedious, since the sign needs to be taken into consideration. In Appendix A, we derive the bounds

$$\begin{aligned} E_1^{(\mathrm{nng})} &\leq \sqrt{\min_{\kappa>0} \left\{(2+2\kappa)\Big[s\ln\left(\tfrac{en}{s}\right) + s\ln\left(\tfrac{1}{2}\right) + s\ln\left(1+\sqrt{1+\tfrac{1}{\kappa}}\right)\Big]\right\} + \tfrac{1}{2}s} \\ &\quad + t\sqrt{s}, \qquad (4.10) \\ E_2^{(\mathrm{nng})} &\leq \Big[n(n-s)\cdot\Big(\big(1+t^2\big)\big(\Phi(-t)-\tfrac{1}{2}\Phi(-t)^2\big) - t\varphi(-t)\Phi(t) \\ &\qquad - \frac{t}{\sqrt{\pi}}\Phi(-t\sqrt{2}) + \frac{1}{2\sqrt{2\pi}}\varphi(-t\sqrt{2})\Big)\Big]^{1/2}. \qquad (4.11) \end{aligned}$$

Hence, we obtain

$$\omega(T_s) \leq \min_{t\geq 0} \left\{E_1^{(\mathrm{nng})} + E_2^{(\mathrm{nng})}\right\}, \qquad (4.12)$$

which leads to the following lower bound on the minimal number of measurements needed for uniform recovery of sparse nonnegative vectors.

**Theorem 4.7.** *Let $A \in \mathbb{R}^{m\times n}$ be a standard Gaussian random matrix, and let $\omega$ be the estimation of $\omega(T_s)$ defined in* (4.12) *with the bounds* (4.10) *and* (4.11)*. If*

$$\frac{m^2}{m+1} \geq \left(\omega + \sqrt{2\ln\left(\tfrac{1}{\varepsilon}\right)}\right)^2,$$

*then every $s$-sparse nonnegative $x \in \mathbb{R}_+^n$ is the unique optimal solution of the nonnegative $\ell_1$-minimization problem* $\min\{\|z\|_1 \,:\, Az = Ax,\ z \geq 0\}$ *with probability at least* $1-\varepsilon$.

The proof of Theorem 4.7 is provided in Appendix A as well.

**Numerical Evaluation and Discussion** Before comparing the result in Theorem 4.7 to the result in Theorem 4.5 for sparse vectors, let us mention that Theorem 4.7 shows that random measurement matrices satisfy the nonnegative null space property ($\mathrm{NSP}_{\geq 0}$) with high probability, given that the number of measurements is sufficiently large. Hence, there exist matrices allowing for uniform recovery of sparse nonnegative vectors with high probability.

Let us now evaluate numerically the computed bound for the minimal number of measurements needed for a Gaussian random matrix to satisfy the nonnegative

NSP. Therefore, we define $\omega^{(\mathrm{lin})}$ to be the bound in (4.7) for the Gaussian width of the set $T_s$ in (4.2) of unit-norm vectors violating the classical NSP, and $\omega^{(\mathrm{nng})}$ to be the bound in (4.12) for the corresponding Gaussian width for sparse nonnegative vectors. For the minimum over $\kappa$ and $t$ in (4.10) and (4.12), we use the values

$$\hat{\kappa} = \sqrt{\frac{n}{n + \ln(m) + n\ln(\frac{1}{2})}} \tag{4.13}$$

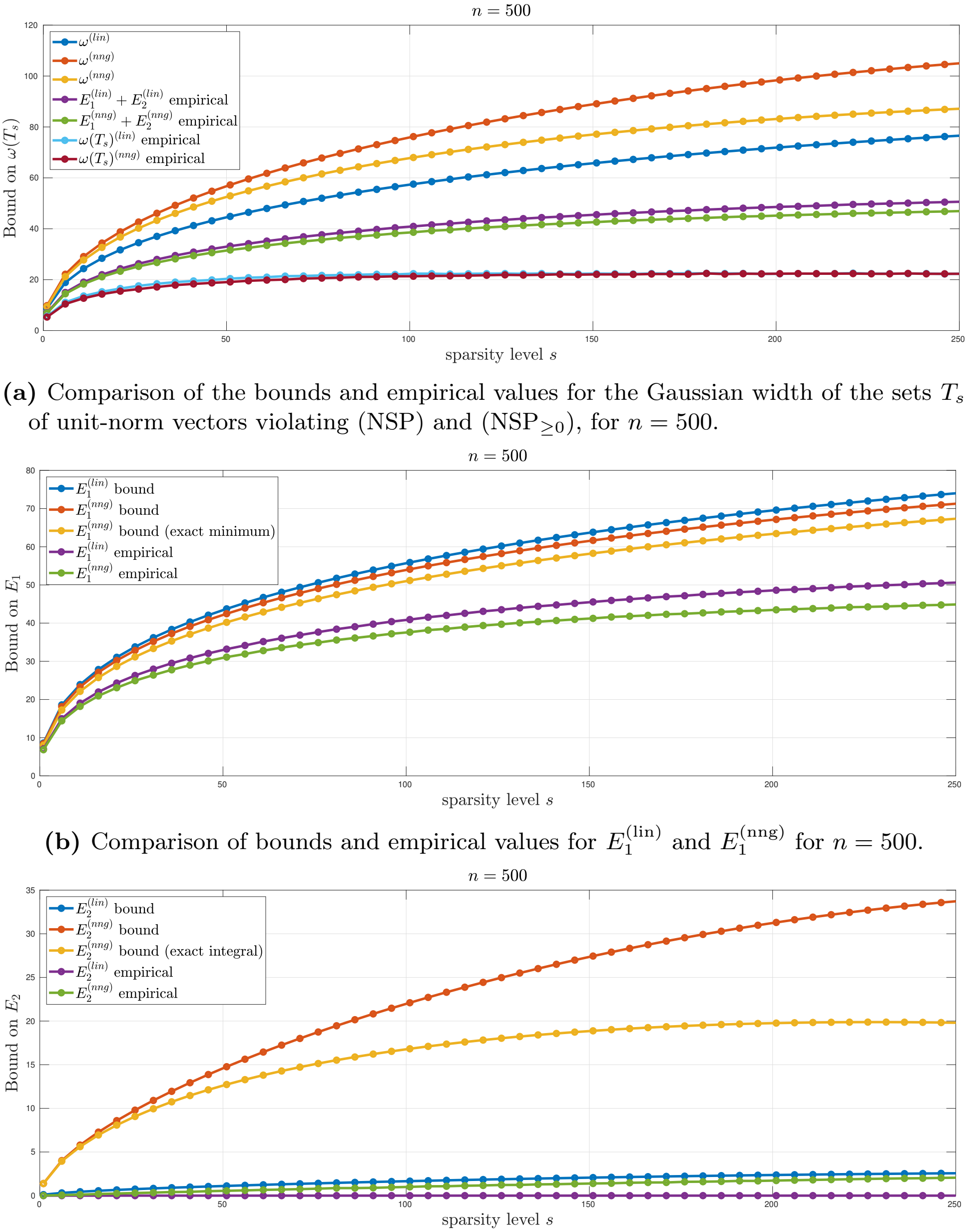
and $\hat{t} = \sqrt{2\ln(en/s)}$, respectively, since, empirically, $\hat{\kappa}$ and $\hat{t}$ are close to the (numerically evaluated) minimum. Moreover, the value $\hat{t}$ is also chosen in the analysis of sparse vectors, and the choice of $\hat{\kappa}$ is justified by the following argument:

$$\begin{aligned} &(2+2\kappa)\Big[\ln(m) + n\ln\big(\tfrac{1}{2}\big) + n\ln\big(1+\sqrt{1+\tfrac{1}{\kappa}}\big)\Big] \\ &\le (2+2\kappa)\Big[\ln(m) + n\ln\big(\tfrac{1}{2}\big) + n\big(\sqrt{1+\tfrac{1}{\kappa}}\big)\Big] \\ &\le (2+2\kappa)\Big[\ln(m) + n\ln\big(\tfrac{1}{2}\big) + n\big(1+\tfrac{1}{\kappa}\big)\Big], \end{aligned} \tag{4.14}$$

and $\hat{\kappa}$ is the minimum of (4.14). Recall from Theorems 4.5 and 4.7 that $\omega^{(\mathrm{lin})}$ and $\omega^{(\mathrm{nng})}$ are a lower bound for the minimal number of random Gaussian measurements needed to satisfy (NSP) and (NSP$_{\geq 0}$), and thus allow for uniform recovery of sparse and sparse nonnegative vectors, respectively.

Figure 4.1a plots $\omega^{(\mathrm{lin})}$ (blue) and $\omega^{(\mathrm{nng})}$ (red and yellow) as a function of the sparsity level for $n = 500$. The latter bound is displayed in two variants: First, directly in the form (4.11) (red), and second, using the numerically exact value for the integral appearing in the derivation of the bound, see (7.6) in Appendix A (yellow). The second variant also uses the numerically evaluated minimal $\kappa$ appearing in the estimation (4.10) of $E_2^{(\mathrm{nng})}$. It turns out that, unfortunately, the derived bound for sparse nonnegative vectors depicted in red is worse than the corresponding bound for general sparse vectors in blue. Using the numerically exact value of the integral in $E_2^{(\mathrm{nng})}$ as well as the numerically evaluated minimal $\kappa$, the derived bound depicted in yellow becomes slightly better, but it is still worse than the blue bound for sparse vectors.

For a comparison, Figure 4.1a also contains empirical values for the sums of the expectations $E_1^{(\mathrm{lin})} + E_2^{(\mathrm{lin})}$ (violet) as well as $E_1^{(\mathrm{nng})} + E_2^{(\mathrm{nng})}$ (green). These have been obtained by sampling 1000 Gaussian random vectors $g$, computing the quantity within the expectation in (4.5), (4.6) as well as (4.9), respectively, and taking the empirical mean of the results. First of all, these results show that both the bound for the linear and the nonnegative case are not precise and rather far from being optimal. Furthermore, empirically, there is a clear difference between the two



**(a)** Comparison of the bounds and empirical values for the Gaussian width of the sets $T_s$ of unit-norm vectors violating (NSP) and (NSP$_{\geq 0}$), for $n = 500$.

**(b)** Comparison of bounds and empirical values for $E_1^{(\mathrm{lin})}$ and $E_1^{(\mathrm{nng})}$ for $n = 500$.

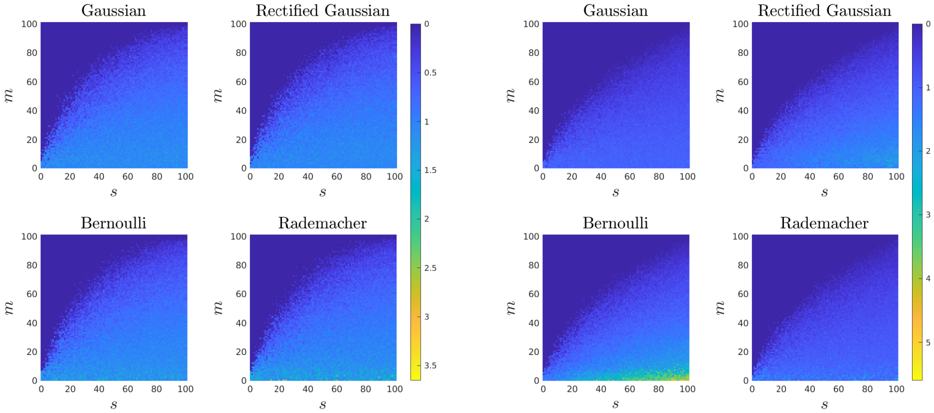
**(c)** Comparison of bounds and empirical values for $E_2^{(\mathrm{lin})}$ and $E_2^{(\mathrm{nng})}$ for $n = 500$.

**Figure 4.1.** Comparison of the bounds for the minimal number of measurements needed for uniform recovery of sparse vectors and sparse nonnegative vectors, for $n = 500$.

bounds and this time, the green bound for sparse nonnegative vectors is smaller than the violet bound for sparse vectors. Thus, empirically, under the additional side constraint $x \geq 0$, fewer measurements are needed for guaranteeing uniform recovery via the nonnegative NSP. Finally, Figure 4.1a displays results for directly simulating the Gaussian widths of the sets $T_s$ for sparse (light blue) and sparse nonnegative vectors (dark red), respectively, see (4.2) and (4.8). These results have been obtained by randomly generating 100 Gaussian random vectors, solving the convex optimization problem $\max\{\langle g, x\rangle \ : \ \|x\|_2 \leq 1, \ x \in T_s\}$ and taking the empirical mean of the solutions. The results show again that the Gaussian width for sparse nonnegative vectors depicted in light blue is smaller than the Gaussian width for sparse vectors depicted in dark red, at least for small sparsity levels $s$.

Figures 4.1b and 4.1c contain a separate comparison of the bounds on $E_1^{(\mathrm{lin})}$ (blue), and $E_1^{(\mathrm{nng})}$ (red, yellow) as well as $E_2^{(\mathrm{lin})}$ (blue), and $E_2^{(\mathrm{nng})}$ (red, yellow), see (4.5), (4.6), (4.10) and (4.11), respectively. We again add an empirical simulation of the respective expectations (violet, green), which are obtained as described above. For $n = 500$, the resulting bounds are plotted as a function of $s$. In case of $E_1^{(\mathrm{nng})}$, the bound is plotted once using $\hat{\kappa}$ as defined in (4.13) (red) and once using the numerically evaluated minimum over $\kappa$ (yellow). The bound for $E_2^{(\mathrm{nng})}$ is also displayed in two variants: First, directly in the form (4.11) (red), and second, using the numerically exact value for the integral appearing in the derivation of the bound, see (7.6) in Appendix A (yellow). As can be seen, the red bound on $E_1^{(\mathrm{nng})}$ is indeed smaller than the blue bound on $E_1^{(\mathrm{lin})}$, even if $\hat{\kappa}$ is used. However, both the red and the yellow bound on $E_2^{(\mathrm{nng})}$ are significantly larger than the blue bound on $E_2^{(\mathrm{lin})}$. One particular estimation that may be responsible for this difference is the inequality in (7.5), see Appendix A. The same inequality is also used in the estimation of the bound on $E_2^{(\mathrm{lin})}$ in (4.6) (see [104, p. 296]). The difference is, however, that for sparse vectors, the $n - s$ smallest entries of $g$ in absolute value contribute to $E_2^{(\mathrm{lin})}$, whereas for $E_2^{(\mathrm{nng})}$, the smallest entries of $g$ are used. This already shows that one can expect $E_2^{(\mathrm{nng})} > E_2^{(\mathrm{lin})}$, which is also confirmed by the empirical values for $E_2^{(\mathrm{nng})}$ in green and $E_2^{(\mathrm{lin})}$ in violet. Hence, the weak estimate (7.5) carries much more weight in the case of sparse nonnegative vectors, which leads to $\omega^{(\mathrm{nng})}$ being larger than $\omega^{(\mathrm{lin})}$. Consequently, improving on this inequality would most likely result in a major improvement of the overall bound for sparse nonnegative vectors.

Altogether, even if the theoretical bound in Theorem 4.7 for sparse nonnegative vectors is larger than its counterpart in Theorem 4.5 for general sparse vectors, our empirical simulations show that fewer measurements for uniform recovery of sparse nonnegative vectors can be expected, in comparison to general sparse vectors. In order to underline this observation a bit further, Figure 4.2 shows empirical results for individual recovery of sparse and sparse nonnegative vectors in the di-



**(a)** Results for sparse vectors. **(b)** Results for sparse nonnegative vectors.

**Figure 4.2.** Empirical success of individual recovery for sparse and sparse nonnegative vectors and different types of random matrices for $n = 100$. The heatmap shows the normalized recovery error (4.15).

mension $n = 100$ using four different types of random matrices. We present results for individual recovery at this point, since this is more natural for conducting simulations. For each combination of the number of measurements $m \in [100]$ and the sparsity level $s \in [100]$, we have drawn 25 random matrices, and an $s$-sparse Gaussian random vector $x \in \mathbb{R}^{100}$ for each matrix. Then, we solved the recovery problem and computed

$$\frac{\|x - x^*\|_2}{\|x\|_2}, \tag{4.15}$$

where $x^*$ denotes the optimal solution of the recovery problem. This experiment has been repeated for four different types of random matrices, namely Gaussian matrices with $A_{ij} \sim \mathcal{N}(0,1)$, rectified Gaussian matrices with $A_{ij} \sim \mathcal{N}^R(0,1)$, Bernoulli matrices with $A_{ij} \sim \text{Unif}(\{0,1\})$ and Rademacher matrices with $A_{ij} \sim \text{Unif}(\{-1,1\})$, where $\text{Unif}(C)$ denotes the uniform distribution on the set $C$. A rectified Gaussian distribution is obtained from the standard Gaussian distribution by setting all negative elements to zero, i.e., if $X \sim \mathcal{N}(0,1)$, then $Y = \max\{0, X\} \sim \mathcal{N}^R(0,1)$ is a rectified Gaussian random variable. The plots in Figure 4.2 show the resulting values of (4.15) for each type of random matrix for sparse and sparse nonnegative vectors. As can be seen, the normalized error (4.15) is zero with high probability in the top left corner in all cases, and increases for larger sparsity levels $s$ and smaller

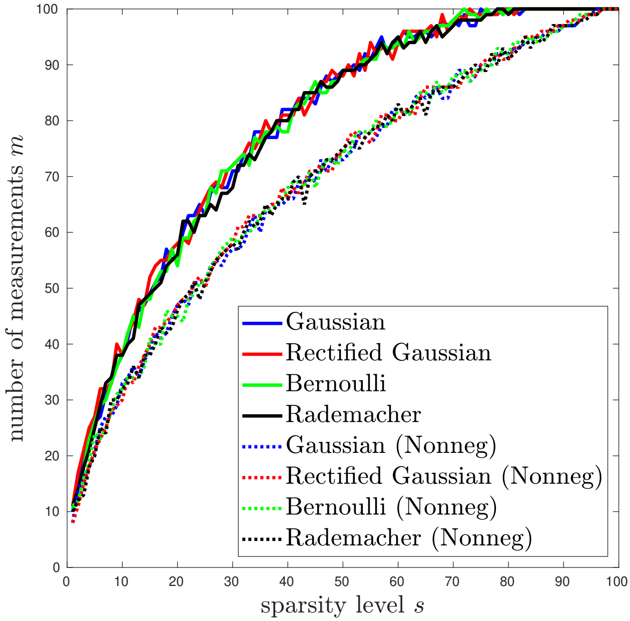


**Figure 4.3.** Comparison of the transition between failure and success for sparse and sparse nonnegative vectors for $n = 100$ and different types of random matrices.

number of measurements $m$. Moreover, the plots show a clear phase transition from failure to success of the recovery process with high probability. It turns out that this transition does not depend on the type of the random matrices, but the phase transition in case of sparse nonnegative vectors occurs for smaller values of $m$, compared to sparse vectors. To underline this point, Figure 4.3 shows the comparison of the phase transitions for sparse and sparse nonnegative vectors. The empirical phase transition for sparse vectors is depicted using solid lines in blue (Gaussian), red (rectified Gaussian), green (Bernoulli) and black (Rademacher), whereas the dashed lines depict the phase transition for sparse nonnegative vectors. These phase transitions are obtained by identifying for each $s$ the minimal value of $m$ such that individual recovery was successful with high probability for all values $m' \geq m$. This comparison reveals that again, for sparse nonnegative vectors fewer measurements are needed for successful individual recovery with high probability, in comparison to sparse vectors.

Thus, both for individual and uniform recovery, explicitly exploiting the side constraint $x \geq 0$ in the recovery process by using nonnegative $\ell_1$-minimization has a positive impact by reducing the number of measurements, i.e., rows of the measurement matrix in order to guarantee successful recovery. In the next chapter, we will consider approaches to testing whether a given measurement matrix satisfies the linear or nonnegative NSP, which guarantees uniform recovery. There, we will also present empirical results for Gaussian random matrices to satisfy these NSPs. These results are the counterpart of the results in Figures 4.2a and 4.2b, since the NSPs characterize uniform recovery. However, computations to test the (nonnegative) NSP are considerably more expensive than solving the (nonnegative) $\ell_1$-minimization problem for a given measurement matrix and sparse (nonnegative) vector. Thus, we can only use the small value $n = 20$. Nevertheless, the results in Figure 5.1 will confirm the results obtained in this chapter, by showing again that there is a difference between sparse and sparse nonnegative vectors in the number of measurements needed to guarantee uniform recovery.

## 4.3 Analysis of Random Measurements for Block-Sparse Matrices

In this section, we will analyze the recovery of block-diagonal matrices under random measurements. We use the null space properties from Section 3.1 to derive bounds for the minimal number of measurements needed to guarantee uniform recovery with high probability. In order to simplify the analysis, we assume symmetric matrices and that the block-structure in $X$ consists of $k$ blocks of equal size $d_1 \times d_2$. Again, we will use Gordon's Escape Theorem 4.4. Note that this formulation holds for the vector space $\mathbb{R}^n$, but it can be easily adapted to the matrix space $\mathbb{R}^{m\times n}$ as well, see, e.g., Kabanava et al. [139] and also the recent overview by Fuchs et al. [108, Theorem 2.1].

**Theorem 4.8** (Gordon's Escape Through a Mesh for $\mathbb{R}^{m\times n}$, [108, Theorem 2.1])**.** *Let $A \in \mathbb{R}^{m\times n}$ be a Gaussian random measurement operator as defined in Section 3.1.1, and let $T$ be a subset of the (Frobenius) unit sphere $\mathbb{S}(\mathbb{R}^{m\times n})$ in $\mathbb{R}^{m\times n}$. Then, for all $t > 0$*

$$\mathbb{P}\Big(\inf_{X\in T} \|A(X)\|_2 \le \sqrt{m-1} - \omega(T) - t\Big) \le \exp\Big(-\tfrac{1}{2}t^2\Big).$$

Using the Frobenius inner product defined in (1.1), the definition of the Gaussian width in Definition 4.2 can be adapted accordingly to

$$\omega(S) := \mathbb{E}\big[\sup\{\langle G, Z\rangle_{\mathrm{F}} \,:\, Z \in S\}\big],$$

where $S \subseteq \mathbb{R}^{m\times n}$ and the expectation is taken over standard Gaussian random matrices $G \in \mathbb{R}^{m\times n}$. By Theorem 3.6, the following null space property of order $s$

characterizes uniform recovery of all $s$-block-sparse block-diagonal matrices:

$$\sum_{i\in S}\|V_{B_i}\|_* < \sum_{i\in \overline{S}}\|V_{B_i}\|_* \tag{NSP$^*_{*,1}$}$$

for all $V \in (\mathrm{null}(A) \cap \mathcal{S}^n) \setminus \{0\}$ and all $S \subseteq [k]$, $|S| \leq s$. For a block-structured matrix $X$ let $\tilde{X}$ be the nonincreasing block-rearrangement of $X$, that is, let $\tilde{X}$ be the matrix with the blocks $X_{B_i}$ of $X$ reordered such that $\|\tilde{X}_{B_1}\|_* \geq \cdots \geq \|\tilde{X}_{B_k}\|_*$ and $X_{B_i} = \tilde{X}_{B_{[\tau(i)]}}$ for all $i \in [k]$ and a permutation $\tau$ of $[k]$ . Then, (NSP$^*_{*,1}$) can be reformulated as

$$\sum_{i=1}^{s}\|\tilde{V}_{B_i}\|_* < \sum_{i=s+1}^{k}\|\tilde{V}_{B_i}\|_* \quad \forall\, V \in (\mathrm{null}(A) \cap \mathcal{S}^n) \setminus \{0\}. \tag{4.16}$$

Due to the scaling invariance of this condition, we can scale $V$ to have a unit norm of 1 with respect to the mixed nuclear-$\ell_2$-norm, that is

$$\|V\|_{*,2} = \Big(\sum_{i=1}^{k}\|V_{B_i}\|_*^2\Big)^{1/2} = 1.$$

Thus, the set $T_s$ of unit-nuclear-norm matrices violating the condition (4.16) is given by

$$T_s = \big\{V \in \mathcal{S}^n \text{ block-structured } : \sum_{i=1}^{s}\|\tilde{V}_{B_i}\|_* \geq \sum_{i=s+1}^{k}\|\tilde{V}_{B_i}\|_*,\ \|V\|_{*,2} = 1\big\}. \tag{4.17}$$

Let $K_s$ be the convex hull of $s$-block-sparse matrices with unit norm with respect to the mixed nuclear-$\ell_2$-norm, i.e.,

$$K_s := \mathrm{conv}\{X \in \mathcal{S}^n \text{ block-structured } : \|X\|_{*,2} = 1,\ \|X\|_{*,0} \leq s\}. \tag{4.18}$$

In the following we show that $T_s$ is contained in $2K_s$ and subsequently estimate the Gaussian width of $K_s$. This is a direct adaption of the corresponding statements for sparse vectors [215, Lemma 4.5], block-sparse vectors [228] and low-rank matrices [139, Lemma 3.4].

**Lemma 4.9.** *Let $s \geq 1$. For $T_s$ and $K_s$ as in* (4.17) *and* (4.18)*, respectively, we have $T_s \subseteq 2K_s$.*

*Proof.* Let $V \in T_s$ be a block-structured matrix and let $\tilde{V}$ be the nonincreasing rearrangement of $V$ with respect to the nuclear norms of the blocks. Then,

$$\sum_{i=1}^{s} \|\tilde{V}_{B_i}\|_* \leq \sqrt{s}\Big(\sum_{i=1}^{s} \|\tilde{V}_{B_i}\|_*^2\Big)^{1/2} \leq \sqrt{s}\Big(\sum_{i=1}^{k} \|\tilde{V}_{B_i}\|_*^2\Big)^{1/2} = \sqrt{s}, \tag{4.19}$$

which implies

$$\sum_{i=s+1}^{k} \|\tilde{V}_{B_i}\|_* \leq \sum_{i=1}^{s} \|\tilde{V}_{B_i}\|_* \leq \sqrt{s}.$$

Now suppose that $\|\tilde{V}_{B_j}\|_* > 1/\sqrt{s}$ for some index $j > s$. Due to the nonincreasing ordering of the blocks, we have $\|\tilde{V}_{B_j}\| \leq \|\tilde{V}_{B_i}\|$ for all $i \in [s]$. This yields

$$\sum_{i=1}^{s} \|\tilde{V}_{B_i}\|_* \geq s\|\tilde{V}_{B_j}\|_* > \sqrt{s},$$

which is a contradiction to (4.19). Thus, we have $\|\tilde{V}_{B_j}\|_* \leq 1/\sqrt{s}$ for all $j > s$. This shows that

$$V \in \mathbb{B}_{*,2}^{S} \times \left(\sqrt{s}\mathbb{B}_{*,1}^{\overline{S}} \cap \tfrac{1}{\sqrt{s}}\mathbb{B}_{*,\infty}^{\overline{S}}\right),$$

where $S$ is the set of indices of the $s$ blocks with largest nuclear norm $\|V_{B_i}\|_*$, and

$$\mathbb{B}_{*,p}^{S} := \{V \in \mathbb{R}^{S \times S} \,:\, \|V\|_{*,p} = 1\}, \quad p \in \{1, 2, \infty\}.$$

Here, $\mathbb{R}^{S \times S}$ denotes the space of matrices consisting of blocks indexed by the elements of $S$. By varying the set $S$ over all possible index sets with $s$ elements, we obtain

$$V \in \bigcup_{E \subseteq [k],\, |E| \leq s} \mathbb{B}_{*,2}^{E} \times \left(\sqrt{s}\mathbb{B}_{*,1}^{\overline{E}} \cap \tfrac{1}{\sqrt{s}}\mathbb{B}_{*,\infty}^{\overline{E}}\right) =: W.$$

Clearly, the extreme points of $W$ are those matrices $V$ with $V = V' + V''$ such that $V' \in \mathbb{R}^{E}$, $\|V'\|_{*,2} = 1$ and $V'' \in \mathbb{R}^{\overline{E}}$ has exactly $s$ nonzero blocks, each with nuclear norm $1/\sqrt{s}$ for some $E \subseteq [k]$ with $|E| = s$. Since $K_s$ is a symmetric and convex set with 0 in its interior, there is a norm $\|\cdot\|_K$ such that $K_s$ is the unit-norm ball with respect to $\|\cdot\|_K$, see, e.g., Rockafellar [211, Theorem 15.2]. Thus, the maximum of $\|\cdot\|_K$ over $W$ is attained at an extreme point of $W$, and for any extreme point $V = V' + V''$ of $W$, we have

$$\|V' + V''\|_K \leq \|V'\|_K + \|V''\|_K \leq 1 + 1 = 2,$$

since clearly $V'$, $V'' \in K_s$. This shows $\max\{\|V\|_K : V \in W\} \leq 2$, which in turn implies $T_s \subseteq W \subseteq 2K_s$. □

This result can now be used to estimate the Gaussian width $\omega(T_s)$ of $T_s$. Therefore, let $G \in \mathbb{R}^{n\times n}$ be a block-structured Gaussian random matrix with $k$ blocks of equal size $d_1 \times d_2$. Then, the Frobenius inner product $\langle G, X\rangle_{\mathrm{F}}$ of $G$ and $X \in \mathcal{S}^n$ satisfies the inequality

$$\begin{aligned}\langle G, X\rangle_{\mathrm{F}} &= \sum_{i=1}^{k} \langle G_{B_i}, X_{B_i}\rangle_{\mathrm{F}} \leq \sum_{i=1}^{k} \|G_{B_i}\|_F \|X_{B_i}\|_F \leq \sum_{i=1}^{k} \|G_{B_i}\|_F \|X_{B_i}\|_* \\ &\leq \sum_{i=1}^{k} \max_{i\in[k]} \{\|G_{B_i}\|_F\} \|X_{B_i}\|_* = \|G\|_{F,\infty} \|X\|_{*,1}.\end{aligned}$$

This implies

$$\begin{aligned}\omega(T_s) \leq 2\omega(K_s) &= 2\mathbb{E}\big[\max\{\langle G, X\rangle_{\mathrm{F}} : \|X\|_{*,0} \leq s,\ \|X\|_{*,2} = 1\}\big] \\ &\leq 2\mathbb{E}\big[\max\{\|G\|_{F,\infty}\|X\|_{*,1} : \|X\|_{*,2} = 1,\ \|X\|_{*,0} \leq s\}\big] \\ &\leq 2\mathbb{E}\big[\max\{\|G\|_{F,\infty}\sqrt{s}\|X\|_{*,2} : \|X\|_{*,2} = 1,\ \|X\|_{*,0} \leq s\}\big] \\ &= 2\sqrt{s}\mathbb{E}\big[\|G\|_{F,\infty}\big] \\ &= 2\sqrt{s}\mathbb{E}\big[\max\{\|\mathrm{vec}(G_{B_i})\|_2 : i \in [k]\}\big] \\ &\leq 2\sqrt{s}\Big(\mathbb{E}\big[\max\{\|\mathrm{vec}(G_{B_i})\|_2^2 : i \in [k]\}\big]\Big)^{1/2} \\ &\leq 2\sqrt{s}\big(\sqrt{2\ln(k)} + \sqrt{d_1 \cdot d_2}\big),\end{aligned}$$

where we used the Jensen inequality in Lemma 7.2 in the penultimate inequality, and [104, Proposition 8.2] for the last inequality. This bound only scales with the blocksize $d_1 \times d_2$ and the number of blocks $k$, but not directly in the overall size of the matrix $X$, i.e., in $kd_1$ and $kd_2$, respectively.

In order to show $\mathbb{P}(\inf_{X\in T_s} \|A(X)\|_2 > 0) \geq 1 - \varepsilon$, we use $t = \sqrt{2\ln(\frac{1}{\varepsilon})}$ in Theorem 4.8. It remains to estimate the minimal number of measurements $m$ so that

$$\sqrt{m-1} - \omega(T_s) - \sqrt{2\ln(\tfrac{1}{\varepsilon})} \geq 0.$$

This is true if

$$\sqrt{m-1} \geq 2\sqrt{s}\big(\sqrt{2\ln(k)} + \sqrt{d_1 \cdot d_2}\big) + \sqrt{2\ln(\tfrac{1}{\varepsilon})},$$

which yields the lower bound

$$m \geq \Big(2\sqrt{s}\big(\sqrt{2\ln(k)} + \sqrt{d_1 \cdot d_2}\big) + \sqrt{2\ln(\tfrac{1}{\varepsilon})}\Big)^2 + 1.$$

Putting the derivation above together, we have proved the following.

**Theorem 4.10.** *Let* $A\colon \mathcal{S}^n \to \mathbb{R}^m$ *be a Gaussian random measurement operator. If*

$$m \geq \Big(2\sqrt{s}\big(\sqrt{2\ln(k)} + \sqrt{d_1 \cdot d_2}\big) + \sqrt{2\ln(\tfrac{1}{\varepsilon})}\Big)^2 + 1,$$

*then, with probability at least* $1-\varepsilon$*, every* $s$*-block-sparse matrix* $X$ *is the unique optimal solution of* $\min\{\|Z\|_{*,1} \,:\, A(Z) = A(X),\ Z \in \mathcal{S}^n\}$.

The lower bound on the number of measurements in Theorem 4.10 only scales with the number of blocks $k$ and the dimension of the blocks $d_1 \cdot d_2$ separately, but not in the overall dimension $k \cdot d_1 \cdot d_2$ of the matrices. This is in contrast to regular sparsity, where the corresponding lower bound scales in the dimension $n$ of the sparse vectors, see Theorem 4.5. Thus, the lower bound takes the specific block-structure into account.

Of course, it also possible to use a similar approach as for sparse nonnegative vectors in Section 4.2. This would require to transform the set $T_s$ defined in (4.17) into a convex cone and find (an inner approximation) of its dual cone. Similarly to the case of nonnegative vectors, we can use the convex cone

$$K_s = \{V \in \mathcal{S}^n \,:\, \sum_{i=1}^{s} \|V_{B_i}\|_* \geq \sum_{i=s+1}^{k} \|V_{B_i}\|_*\}.$$

Since only the nuclear norm of the blocks $V_{B_i}$ but not the entries of the blocks themselves appear in the NSP condition, the computation can in principle be reduced to the case of vectors, where the entries of the vector $v$ are the nuclear norms of the blocks in $V$. Similarly, the Gaussian random matrix reduces to a random vector, whose entries are no longer standard Gaussian random variables. By using the emerging distribution instead of the standard Gaussian distribution, the same proof idea as in the case of vectors could in principle be applied, see [104, Theorem 9.29].

The recovery for positive semidefinite block-structured matrices under random Gaussian measurements is unfortunately considerably harder to analyze. Recall that the corresponding null space property ($\mathrm{NSP}^*_{*,1,\succeq 0}$) in Definition 3.3 reads

$$V_{B_i} \preceq 0 \,\forall\, i \in \overline{S} \quad\implies\quad \sum_{i\in S} \mathbb{1}^\top \lambda(V_{B_i}) < \sum_{i\in\overline{S}} \|V_{B_i}\|_* \tag{4.20}$$

for all $V \in (\mathrm{null}(A) \cap \mathcal{S}^n)\backslash\{0\}$ and all $S \subseteq [k]$, $|S| \leq s$, where $\lambda(V_{B_i})$ is the vector of eigenvalues of $V_{B_i}$. Clearly, this condition is also invariant under permutation of the blocks. However, in contrast to the NSP for block-structured matrices in ($\mathrm{NSP}^*_{*,1}$), not only the nuclear norm of the blocks but also the signs of the eigenvalues appear in the NSP condition in (4.20). Thus, it is not clear in which way the blocks in $V$ need to be ordered so that the set $S$ can be eliminated from the condition, since this ordering necessarily has to take the signs of the eigenvalues within the blocks into consideration. A natural ordering would be lexicographically sorting the blocks with respect to the largest eigenvalue. However, this ordering cannot be used to maximize the inner product $\langle G, V\rangle_{\mathrm{F}}$ over the set of matrices $V$ violating the NSP (4.20), as the following toy example demonstrates. Consider two block-structured matrices $G$ and $V \in \mathcal{S}^4$ with eigenvalues $\lambda(G) = ([2,-1],[6,-20])^\top$ and $\lambda(V) = ([10,10],[-1,-1])^\top$. Sorting the matrices blockwise with respect to the largest eigenvalue yields $\lambda(\hat{G}) = ([6,-20],[2,-1])^\top$ and $\lambda(\hat{V}) = ([10,10,],[-1,-1])^\top$. However, $\langle G, V\rangle_{\mathrm{F}} = 24 > -124 = \langle \hat{G}, \hat{V}\rangle_{\mathrm{F}}$, so that the sorting does not increase the inner product. Consequently, abstracting to the eigenvalues of the blocks and then using an approach to analyze the NSP under Gaussian random measurements similar to the case of nonnegative vectors in Section 4.2 can only be applied if the correct ordering has been identified. This also implies that even simulating the Gaussian width becomes a complicated optimization problem.

Moreover, as already mentioned in the case of sparse nonnegative vectors, a simple approach using an embedding of the set of matrices violating the NSP into a convex set of sparse unit-norm matrices as in Lemma 4.9 is also not possible, since the correct convex set $K_s$ has not yet been identified.

## 4.4 Concluding Remarks and Outlook

The empirical evaluation of the Gaussian width of the set of vectors violating the linear and the nonnegative NSP in Figure 4.1a and the results for individual recovery of sparse and sparse nonnegative vectors in Figures 4.2, 4.2a and 4.2b indicates that fewer measurements seem to be needed for the uniform recovery of sparse nonnegative vectors in comparison to uniform recovery of all sparse vectors. However, the numerical evaluation in Figure 4.1a of the bound for the minimal number of measurements needed for uniform recovery of sparse nonnegative vectors derived in Theorem 4.7 shows that this bound is worse than the corresponding bound for sparse vectors. Thus, it remains an open question to improve on the bound in Theorem 4.7. As outlined in the discussion in Section 4.2, especially one inequality seems to leave much room for improvement. A more precise estimation most likely needs more involved results on rectified Gaussian random variables and vectors.

Another major open question is the derivation of bounds for uniform recovery of positive semidefinite block-structured matrices. Since the NSP for positive semidefinite block-structured matrices only depends on the eigenvalues of the blocks, it makes sense to first consider uniform recovery for block-structured nonnegative vectors and then generalize the result to matrices. However, as described above, the correct ordering for the blocks has not yet been identified. Moreover, in the literature, to the best of the author's knowledge, for block-sparse nonnegative vectors only individual recovery has been analyzed under random measurements by Stojnic [229].

Apart from the results for uniform recovery, it would also be interesting to derive results for individual recovery. To do so, the methods developed in [9, 49] could be employed. Note that individual recovery of sparse nonnegative vectors under Gaussian random measurements has been treated in [226], but the case of (positive semidefinite) block-structured matrices has not yet been considered in the literature. These two cases should be easy adaptions of the block-sparse and block-sparse nonnegative case, which has been analyzed in [229, 230]. A generalization of all these considerations would be to prove a statement of recovery under random measurements in the general framework from Chapter 3. As a starting point, the framework of atomic "norms" [49] or decomposable norms [44] could be used and the corresponding statements for individual recovery in these frameworks extended to uniform recovery.

In a different direction of research, it would be interesting to see if the polytope approach from Donoho and Tanner for analyzing recovery under random measurements also works for more settings than sparse (nonnegative) vectors. In the case of matrices, a direct adaption would replace the unit-norm ball of the $\ell_1$-norm by the unit-norm ball of the nuclear norm. This set is known to be a spectrahedron, see Saunderson et al. [216, Theorem 1.2]. However, the projection of a spectrahedron is not necessarily a spectrahedron in general, see Ramana and Goldman [208, Section 3.1], as opposed to polytopes. Moreover, it seems that a block-structure cannot be easily represented using polytopes.

On top of that, analyzing random matrices for the two remaining special cases treated in Chapter 3, namely integrality and constant modulus constraints is also left open for future research. In case of constant modulus constraints, this would require to generalize the probabilistic tools to the complex setting, or, to employ a explicit split into real and imaginary parts. Since these parts cannot be considered, e.g., sorted, independently but a coupling between these two parts need to be maintained, it is not as straight-forward to apply the methodology used within this chapter. For integrality constraints, the corresponding NSP is not even invariant under scaling of the vectors, so that we cannot assume that the set $T_s$ of integral vectors violating the corresponding NSP is bounded and a subset of the unit sphere. Thus, again, the methodology is not applicable, since we cannot operate on a subset of the unit sphere. Instead, we would need to derive bounds for the $\ell_2$-norm of the vectors $x$ in the null space of a (Gaussian) random measurement matrix $A$ with $x \in T_s$. Moreover, even if it is possible to obtain such a bound, the set $T_s$ is still not convex and not even a cone. It would certainly be interesting to see if the general approach using Gordon's Escape Theorem 4.4 can be modified to work in this case as well, or if it is even possible to introduce the integrality constraint into the analysis based on polytope geometry from Donoho and Tanner.

CHAPTER 5

# Computing Recovery Conditions

In the last chapter, we have considered the question whether there exist matrices which can satisfy the NSP conditions for individual and uniform recovery for various special cases presented in Chapter 3. Recall that a measurement matrix $A$ allows for individual recovery, if a *single* fixed $s$-sparse element is successfully recovered by the corresponding recovery program using $A$. In contrast, uniform recovery means that the corresponding recovery program using a fixed $A$ successfully recovers *all* $s$-sparse elements. We have seen that if the number of measurements, is large enough, then a Gaussian random matrix satisfies the corresponding NSP with high probability. In case of sparse vectors, the number of measurements is given by the number of rows of the measurement matrix. Moreover, by exploiting additional side constraints such as nonnegativity or positive semidefiniteness this number of minimal measurements decreases. Thus, from a theoretical point of view, the presented NSP conditions for different special cases, with and without additional side constraints, are meaningful in the sense that they can be satisfied by matrices and that exploiting side constraints has a positive effect. For a thorough analysis of NSP conditions it now remains to consider the question of how to verify that a given matrix satisfies an NSP condition in practice. This will be done in this chapter. First, we shortly present the case of the classical NSP for sparse vectors, which has been treated by d'Aspremont and El Ghaoui [59], where an SDP formulation for testing the NSP of a measurement matrix was proposed. We complement this by deriving two slightly different MIP formulations for testing the NSP. Afterwards, we extend this formulation to the NSPs for recovering sparse nonnegative vectors and block-sparse vectors with and without additional nonnegativity. For the NSP for sparse vectors and the NSP for sparse nonnegative vectors, we present computational results for varying sizes of the measurement matrix and sparsity levels.

For a very small dimension $n = 20$, we consider Gaussian random measurement matrices $A \in \mathbb{R}^{m \times n}$, that is, random matrices, where all entries are independent standard normal random variables. We show empirically that these random matrices satisfy the nonnegative NSP with high probability for more combinations of sparsity level $s$ and number of measurements $m$, in comparison to the NSP for recovery of sparse vectors. This supplements the consideration in the previous chapter, where a bound for the minimal number of measurements needed for a Gaussian random matrix to satisfy the nonnegative NSP was derived. The numerical evaluations, and especially the empirical comparison of individual recovery for sparse (nonnegative) vectors, show a difference in the number of measurements needed for successful recovery between sparse vectors and sparse nonnegative vectors. Since the (nonnegative) NSP characterizes uniform recovery of (nonnegative) sparse recovery, the results in this chapter add an empirical comparison of uniform recovery, which confirms the results for individual recovery obtained in the previous chapter. Afterwards, we shortly comment on the NSPs for recovery of low-rank (positive semidefinite) matrices and block-diagonal (positive semidefinite) matrices, for which it does not seem to be as easy to formulate them as a MIP or an MISDP. Currently, only immediate nonlinear formulations are known.

In Section 5.2 we consider another condition which is sufficient for uniform recovery of sparse vectors, the restricted isometry property. This condition can be formulated as an MISDP, which was established by Gally and Pfetsch [111]. Chapter 6 shortly introduces general MISDPs and presents several presolving techniques for general MISDPs, which is based on joint work with Marc E. Pfetsch [174]. Further specialized components that can be exploited when solving the MISDP formulation of the restricted isometry property are then derived in Section 5.3. A numerical evaluation of these components follows at the end of Chapter 6.

## 5.1 A Mixed-Integer Programming Formulation for the Null Space Property

Recall from Example (2.12.1) that the null space property (NSP) for characterizing uniform recovery reads

$$\|v_S\|_1 < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \operatorname{null}(A) \setminus \{0\},\ \forall\, S \subseteq [n],\ |S| \leq s. \tag{NSP}$$

By adding $\|v_s\|_1$ to both sides of the inequality and scaling $\|v\|_1 = 1$, (NSP) can be written as

$$\max\{\|v_S\|_1 \,:\, Av = 0,\ \|v\|_1 \leq 1,\ |S| \leq s\} < \frac{1}{2}, \tag{5.1}$$

which can equivalently be formulated as

$$\max\{y^\top v \,:\, Av = 0,\ \|v\|_1 \leq 1,\ \|y\|_\infty \leq 1,\ \|y\|_1 \leq s\} < \frac{1}{2}, \tag{5.2}$$

see, e.g., [59], which follows from homogeneity and the $\ell_\infty$-norm being the dual norm of the $\ell_1$-norm. The quantity in (5.1) is also known as *null space constant (NSC)*. Recall that checking whether a given matrix satisfies the NSP is $\mathcal{NP}$-hard, see Tillmann and Pfetsch [237]. Thus, not much attention has been paid to the problem of computing the exact NSC for a given matrix. Cho et al. [53] propose an approach to compute the exact NSC based on a branch-and-bound approach. Moreover, lower and upper bounds for the NSC are derived using an SDP relaxation in [59] as well as an LP relaxation in Juditsky and Nemirovski [135]. Bounds for the corresponding NSC for sparse nonnegative vectors are obtained by Juditsky et al. [136].

In the following, it is our goal to formulate the optimization problem (5.2) as a MIP. This allows us to check in practice whether a given measurement matrix $A$ satisfies the NSP and thus admits uniform recovery of sparse vectors. We discuss two slightly different formulations and compare them numerically. In a similar spirit, Tillmann [236] derives a MIP formulation for computing the spark of a matrix. Recall that the spark is a recovery condition for sparse recovery using the $\ell_0$-norm.

To start, any optimal solution $(y^*, v^*)$ of Problem (5.2) has $y_i^* \in \{\pm 1\}$ for exactly $s$ indices $i \in [n]$ and $y_j^* = 0$ otherwise. Thus, $y^*$ selects entries of $v^*$, which then form the set $S$ in (NSP). Consequently, the variables $y_i$ in the objective function can assumed to satisfy $y_i \in \{0, \pm 1\}$ for all $i \in [n]$. Moreover, in any optimal solution $(y^*, v^*)$, the signs of $v_i^*$ and $y_i^*$ coincide for all $i \in [n]$ with $y_i^* \neq 0$. The $\ell_1$-norm constraint on $v$ can be modeled as a linear constraint by using a variable split. For $v \in \mathbb{R}$, define its positive part $v^+$ and negative part $v^-$ as

$$v^+ := \max\{0, v\}, \qquad v^- := \max\{0, -v\}.$$

Clearly, when introducing variables $v_i^+$ and $v_i^-$ instead of $v_i$, we need to ensure that $v_i^+$ and $v_i^-$ are not simultaneously nonzero. Such a constraint which models that for a set of variables $x_1, \ldots, x_n$, at most one variable $x_i \neq 0$, $i \in [n]$, whereas $x_j = 0$ for all $j \neq i$, is called an *sos1-constraint*, which we denote with $\operatorname{sos1}(x_1, \ldots, x_n)$. It can be modeled by adding the set of constraints $x_i \cdot x_j = 0$ for all $i \neq j$. In the presence of bounds, an sos1-constraint can be linearized by adding additional binary variables. Assume that $-\infty < \ell_i \leq x_i \leq u_i < \infty$ holds for all $i \in [n]$, and let $d_i \in \{0,1\}$ for $i \in [n]$ be binary variables. Then, the constraint $\operatorname{sos1}(x_1, \dots, x_n)$ can be modeled by adding the following constraints:

$$\sum_{i=1}^{n} d_i \leq 1, \qquad \ell_i\, d_i \leq x_i \leq u_i\, d_i \quad \forall\, i \in [n], \qquad d_i \in \{0,1\} \quad \forall\, i \in [n]. \tag{5.3}$$

Using a variable split into positive and negative part together with an sos1-constraint, the constraint $\|v\|_1 \leq 1$ can be equivalently formulated as

$$\sum_{i=1}^{n} \big(v_i^+ + v_i^-\big) \leq 1, \qquad \operatorname{sos1}(v_i^+, v_i^-) \quad \forall\, i \in [n], \qquad v_i^+,\, v_i^- \in [0,1] \quad \forall\, i \in [n].$$

Due to the simple bounds $v_i^+$, $v_i^- \in [0,1]$, Inequalities (5.3) for modeling the constraint $\operatorname{sos1}(v_i^+, v_i^-)$ simplify to

$$v_i^{\pm} \leq d_i^{\pm}, \quad d_i^+ + d_i^- \leq 1, \quad d_i^{\pm} \in \{0,1\} \quad \forall\, i \in [n]. \tag{5.4}$$

As already mentioned, every optimal solution $(y^*, v^*)$ of Problem (5.2) satisfies $y_i^* \in \{0, \pm 1\}$, and if $y_i^* \neq 0$, then the signs of $y_i^*$ and $v_i^*$ coincide. Thus, when using the variable split $v_i = v_i^+ - v_i^-$, we can assume $y_i$ to be binary and write the objective function as $\sum_{i=1}^n \big(y_i\, v_i^+ + y_i\, v_i^-\big)$. In order to linearize the bilinear terms in the objective function, we use the standard McCormick relaxation [176]. To do so, we introduce new variables $w_i^{\pm} \in [0,1]$ which replace the bilinear terms $y_i v_i^{\pm}$ for $i \in [n]$ and add the inequalities

$$w_i^{\pm} \leq y_i, \quad w_i^{\pm} \leq v_i^{\pm}, \quad w_i^{\pm} \geq y_i + v_i^{\pm} - 1$$

for all $i \in [n]$. Since the sum $\sum_{i=1}^n (w_i^+ + w_i^-)$ is maximized in the objective function, the last set of inequalities $w_i^{\pm} \geq y_i + v_i^{\pm} - 1$ can be omitted, and the obtained relaxation is in fact exact, that is, if $y_i = 1$, then $w_i^{\pm} = v_i^{\pm}$ and $y_i = 0$ implies $w_i^{\pm} = 0$. Overall, this leads to the following MIP formulation of (5.2):

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} \big(w_i^+ + w_i^-\big) \\
\text{s.t.} \quad & A(v^+ - v^-) = 0, \qquad \sum_{i=1}^{n} \big(v_i^+ + v_i^-\big) \leq 1, \qquad \sum_{i=1}^{n} y_i \leq s, \\
& w_i^{\pm} \leq y_i, \qquad w_i^{\pm} \leq v_i^{\pm}, \qquad v_i^{\pm} \leq d_i^{\pm} \qquad d_i^+ + d_i^- \leq 1 \qquad \forall\, i \in [n], \\
& y_i \in \{0,1\}, \quad d_i^{\pm} \in \{0,1\}, \quad w_i^{\pm},\, v_i^{\pm} \in [0,1] \qquad \forall\, i \in [n].
\end{aligned} \tag{5.5}$$

In this formulation, the sos1-constraint on $v_i^+$, $v_i^-$ (or, to be more precise, its linearization using the binary variables $d_i^\pm$) cannot be omitted, since otherwise setting $v_1^\pm = w_1^\pm = \frac{1}{2}$, $y_1 = 1$, and setting all remaining variables to zero yields a feasible solution with an objective value of 1, independent of the choice of $A$. This is due to the fact that the linearized objective function $\sum_{i=1}^n (w_i^+ + w_i^-) = \sum_{i=1}^n y_i \cdot (v_i^+ + v_i^-)$ is equal to $y^\top v$ only if the sos1-constraints on $v_i^\pm$ are satisfied.

However, it is possible to modify this formulation, so that the sos1-constraints become superfluous. This can be achieved by splitting the variables $y_i$ also into a positive and negative part $y_i^\pm$, and using the objective function

$$\sum_{i=1}^n \bigl(y_i^+ v_i^+ + y_i^- v_i^- - y_i^+ v_i^- - y_i^- v_i^+\bigr).$$

By introducing auxiliary variables $w_i^{(1)} = y_i^+ v_i^+$, $w_i^{(2)} = y_i^- v_i^-$, $w_i^{(3)} = y_i^+ v_i^-$ and $w_i^{(4)} = y_i^- v_i^+$, we obtain the following alternative formulation

$$\begin{aligned}
\max \quad & \sum_{i=1}^n \bigl(w_i^{(1)} + w_i^{(2)} - w_i^{(3)} - w_i^{(4)}\bigr) && && \text{(5.6a)}\\
\text{s.t.} \quad & A(v^+ - v^-) = 0, \quad \sum_{i=1}^n \bigl(v_i^+ + v_i^-\bigr) \le 1, \quad \sum_{i=1}^n \bigl(y_i^+ + y_i^-\bigr) \le s, && && \text{(5.6b)}\\
& w_i^{(1)} \le y_i^+, \quad w_i^{(2)} \le y_i^-, \quad w_i^{(3)} \le y_i^+, \quad w_i^{(4)} \le y_i^- && \forall\, i \in [n], && \text{(5.6c)}\\
& -1 + y_i^+ + w_i^{(1)} \le v_i^+ \le w_i^{(1)} + 1 - y_i^+ && \forall\, i \in [n], && \text{(5.6d)}\\
& -1 + y_i^- + w_i^{(2)} \le v_i^- \le w_i^{(2)} + 1 - y_i^- && \forall\, i \in [n], && \text{(5.6e)}\\
& -1 + y_i^+ + w_i^{(3)} \le v_i^- \le w_i^{(3)} + 1 - y_i^+ && \forall\, i \in [n], && \text{(5.6f)}\\
& -1 + y_i^- + w_i^{(4)} \le v_i^+ \le w_i^{(4)} + 1 - y_i^- && \forall\, i \in [n], && \text{(5.6g)}\\
& y_i^\pm \in \{0,1\}, \quad w_i^{(1)}, w_i^{(2)}, w_i^{(3)}, w_i^{(4)}, v_i^\pm \in [0,1] && \forall\, i \in [n], && \text{(5.6h)}
\end{aligned}$$

which does not need the sos1-constraints $\mathrm{sos1}(v_i^+, v_i^-)$ and $\mathrm{sos1}(y_i^+, y_i^-)$ for all $i \in [n]$, as proven in the next lemma.

**Lemma 5.1.** *There always exists an optimal solution of Problem* (5.6) *which satisfies the sos1-constraints*

$$\textit{sos1}(v_i^+, v_i^-), \quad \textit{sos1}(y_i^+, y_i^-) \qquad \forall\, i \in [n]. \tag{5.7}$$

*Proof.* Let $(\hat{y}^\pm, \hat{w}^{(1)}, \hat{w}^{(2)}, \hat{w}^{(3)}, \hat{w}^{(4)}, \hat{v}^\pm)$ be an optimal solution of Problem (5.6) so that there exists an index $i \in [n]$ with $\hat{v}_i^+$, $\hat{v}_i^- > 0$. Without loss of generality we assume $\hat{v}_i^+ \geq \hat{v}_i^-$. In order to show that there always exists another optimal solution with the same objective value which satisfies the sos1-constraints (5.7), we distinguish between different cases for the value of the binary variables $\hat{y}_i^\pm$.

If $\hat{y}_i^+ = \hat{y}_i^- = 0$, then Constraint (5.6c) implies $\hat{w}_i^{(1)} = \hat{w}_i^{(2)} = \hat{w}_i^{(3)} = \hat{w}_i^{(4)} = 0$. The constraint $\sum_i(\hat{v}_i^+ + \hat{v}_i^-) \leq 1$ in (5.6b) implies that setting $\tilde{v}_i^+ := \hat{v}_i^+ - \hat{v}_i^- \geq 0$ and $\tilde{v}_i^- := 0$ is also feasible and does not change the objective value. Furthermore, $\tilde{v}^\pm$ clearly satisfies the sos1-constraints (5.7).

If $\hat{y}_i^+ = 1$ and $\hat{y}_i^- = 0$, then $\hat{w}_i^{(1)} = \hat{v}_i^+ > 0$ by Constraint (5.6d) and $\hat{w}_i^{(3)} = \hat{v}_i^- > 0$ by Constraint (5.6f). Thus, the term $\hat{w}_i^{(1)} - \hat{w}_i^{(3)} = \hat{v}_i^+ - \hat{v}_i^-$ is contained in the objective function. Again, $\tilde{v}_i^+ := \hat{v}_i^+ - \hat{v}_i^- \geq 0$ and $\tilde{v}_i^- := 0$ is also feasible and does not change the objective value, since $\hat{w}_i^{(1)} - \hat{w}_i^{(3)} = \tilde{v}_i^+ - \tilde{v}_i^- = \hat{v}_i^+ - \hat{v}_i^-$ in this case. Again, the sos1-constraints (5.7) are satisfied by $\tilde{v}^\pm$. The same holds for the case $\hat{y}_i^- \neq 0$ and $\hat{y}_i^+ = 0$.

Lastly, if $\hat{y}_i^+ = \hat{y}_i^- = 1$, then $\hat{w}_i^{(1)} = \hat{w}_i^{(4)} = \hat{v}_i^+ > 0$ and $\hat{w}_i^{(2)} = \hat{w}_i^{(3)} = \hat{v}_i^- > 0$, so that the contribution of index $i$ to the objective function is 0. This implies that defining $\tilde{v}_i^+ := \hat{v}_i^+ - \hat{v}_i^- \geq 0$, $\tilde{v}_i^- = 0$, as well as $\tilde{y}_i^+ := 1$, $\tilde{y}_i^- = 0$ and setting $\tilde{w}_i^{(j)}$ accordingly yields a feasible solution which strictly increases the objective value, which is a contradiction to the optimality of $(\hat{y}^\pm, \hat{w}^{(1)}, \hat{w}^{(2)}, \hat{w}^{(3)}, \hat{w}^{(4)}, \hat{v}^\pm)$, so that this case cannot occur.

Overall, there always exists an optimal solution of the MIP formulation (5.6) with the same objective value which satisfies the sos1-constraints on $v_i^\pm$ for all $i \in [n]$. □

As another approach to circumvent the bilinear term $y^\top v$ in the objective function of Problem (5.2), d'Aspremont and El Ghaoui [59] use a change of variables $V = vv^\top$, $Y = yy^\top$ and $Z = yv^\top$ to lift the problem into a higher-dimensional space. Using that $X = xx^\top$ if and only if $X \succeq 0$ and $\operatorname{rank}(X) = 1$ for a matrix $X$ and a vector $x$, Problem (5.2) becomes the SDP

$$\max \quad \operatorname{tr}(Z) \tag{5.8a}$$

$$\text{s.t.} \quad AVA^\top = 0, \quad \|V\|_1 \leq 1, \quad \|Y\|_\infty \leq 1, \quad \|Y\|_1 \leq s^2 \quad \|Z\|_1 \leq s, \tag{5.8b}$$

$$\begin{pmatrix} V & Z^\top \\ Z & Y \end{pmatrix} \succeq 0, \quad \operatorname{rank}\begin{pmatrix} V & Z^\top \\ Z & Y \end{pmatrix} = 1, \tag{5.8c}$$

$$V,\, Y \in \mathcal{S}^n,\ Z \in \mathbb{R}^{n\times n}. \tag{5.8d}$$

The rank-constraint in (5.8c) ensures that in an optimal solution of Problem (5.8), we have $V = vv^\top$, $Y = yy^\top$ and $Z = yv^\top$. The norms in (5.8b) are to be understood entrywise, i.e., $\|Y\|_\infty = \max\{|Y_{ij}| \,:\, 1 \leq i, j \leq n\}$ and $\|Y\|_1 = \sum_{i,j}|Y_{ij}|$. Dropping the rank-constraint yields an SDP relaxation. Importantly, the constraint $\|Z\|_1 \leq s$ is redundant only in the rank-constrained SDP (5.8), but not in the SDP relaxation

**Table 5.1.** Results for the MIP formulation of the linear NSP on a testset of 100 Gaussian random matrices.

| formulation | #opt | #nodes | time |
|---|---:|---:|---:|
| MIP (5.5) (linearized sos1) | 16 | 4 190 605.8 | 2686.74 |
| MIP (5.10) (explicit sos1) | 22 | 3 291 349.2 | 2278.98 |
| MIP (5.6) (without sos1) | 100 | 27 355.8 | 36.43 |

without the rank-constraint. In [59], the tightness and performance of the SDP relaxation is analyzed theoretically and numerically.

In order to evaluate the performance of the proposed MIP formulations (5.5) and (5.6), we generate 100 standard Gaussian random matrices $A \in \mathbb{R}^{m \times n}$ with $A_{ij} \sim \mathcal{N}(0,1)$. Namely, there are five matrices per combination $(n, m, s)$ with

$$n \in \{20, 40, 60, 80, 100\}, \quad m \in \{7, 15\}, \quad s \in \{2, 3\}. \tag{5.9}$$

For each random matrix, we use three formulations for testing the NSP. First, we solve (5.5), where the sos1-constraints on $v_i^{\pm}$, $i \in [n]$ are linearized. Then, we solve (5.5) where the sos1-constraints on $v_i^{\pm}$, $i \in [n]$ are directly added as sos1-constraint without a linearization, i.e.,

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} \left(w_i^+ + w_i^-\right) \\
\text{s.t.} \quad & A(v^+ - v^-) = 0, \qquad \sum_{i=1}^{n} \left(v_i^+ + v_i^-\right) \leq 1, \qquad \sum_{i=1}^{n} y_i \leq s, \\
& w_i^{\pm} \leq y_i, \qquad w_i^{\pm} \leq v_i^{\pm}, \qquad \mathrm{sos1}(v_i^+, v_i^-) \qquad\qquad \forall\, i \in [n], \\
& y_i \in \{0,1\}, \quad w_i^{\pm},\, v_i^{\pm} \in [0,1] \qquad\qquad\qquad\qquad \forall\, i \in [n].
\end{aligned} \tag{5.10}$$

The sos1-constraints can be handled by, e.g., using methods by Fischer and Pfetsch [96]. For a brief description of the handling of sos1-constraints in the solver SCIP [219], see the corresponding section in the release report of SCIP 3.2 [113].

Lastly, we solve (5.6), which does not need any sos1-constraints. For our computations, we use SCIP 7.0.4 [114] with SoPlex 5.0.2 as LP solver. All tests were performed on a Linux cluster with 3.5 GHz Intel Xeon E5-1620 Quad-Core CPUs, having 32 GB main memory and 10 MB cache. All computations were run single-threaded and with a time limit of one hour. Table 5.1 shows the number of instances solved to optimality as well as the shifted geometric means of the number of processed nodes and the solution time in seconds, with a shift of 100 nodes and 1 second. Using the formulation (5.5) with linearized sos1-constraints can only solve 16 out of the 100 instances within the time limit, namely only those instances with $(n, m, s) \in \{(20, 7, 2), (20, 15, 2), (20, 15, 3)\}$ and one of the five instances of type $(n, m, s) = (20, 7, 3)$. For all other instances, at least a nontrivial primal bound greater than 0 is found. Note that if this primal bound is already larger than $\frac{1}{2}$, this suffices to show that the NSP is violated.

If the sos1-constraints are added as explicit constraints to SCIP, 22 instances can be solved within the time limit, namely all 20 instances with $n = 20$ and two of the five instances with $(n, m, s) = (40, 7, 2)$. Again, for all other instances, at least a nontrivial primal bound is found. Using the formulation (5.6) which does not need sos1-constraint clearly outperforms both sos1-based formulations by almost two orders of magnitude in terms of the solution time as well as the number of used nodes. Moreover, all 100 instances are solved to optimality. This shows that introducing additional (continuous) variables is very beneficial because it allows to significantly reduce the number of needed integral variables, since no sos1-constraints are needed.

Overall, seven matrices satisfy the NSP, namely all matrices with parameters $(n, m, s) = (20, 15, 2)$, one matrix with $(n, m, s) = (20, 15, 3)$, and one matrix with $(n, m, s) = (40, 15, 2)$. Additionally, we also tested the SDP formulation (5.8). Since the exact formulation with the rank1-constraint cannot be solved for any of the sizes in (5.9), we omit the rank1-constraint and compare the bound of the resulting relaxation with the optimal solution computed with the MIP formulation (5.6). Since these results were run in Matlab using cvx [122], and on a different computer than the experiments with the MIP formulations, we do not report or compare the solution times. Within a time limit of one hour, 99 of the 100 instances could be solved to optimality. It turns out that for all instances, the SDP relaxation indeed produces a larger optimal solution than the exact MIP formulation. However, for five of the matrices which satisfy the NSP, the SDP-bound is smaller than $\frac{1}{2}$, which also suffices to show that the NSP holds.

If the number of rows and the sparsity levels are increased, the MIPs become much harder to solve. For each of the 20 combinations of $(n, m, s)$ as depicted in Table 5.2, we also create five Gaussian random matrices, and solve the three different MIP formulations with the same setup as before. Table 5.3 shows the number of solved instances and the shifted geometric means of the number of used nodes and the solution times. It turns out that only increasing the number of rows and/or the sparsity level already leads to instances which cannot be solved by the MIP formulation (5.6) anymore. Only the instances with $n = 20$, those with $(n, m) = (40, 15)$, $(n, m, s) = (80, 20, 3)$ and two of the instances with $(n, m, s) = (100, 30, 3)$ could be solved to optimality within a time limit of one hour. The other two formulations (5.5) and (5.10) even failed to solve all in-

**Table 5.2.** Sparsity levels and sizes of the random matrices used for evaluating the MIP formulation of the linear and nonnegative NSP.

| #cols $n$ | #rows $m$ | sparsity $s$ |
|---|---|---|
| 20 | 7 | $\{2, 3\}$ |
| | 15 | $\{3, 5\}$ |
| 40 | 15 | $\{3, 4\}$ |
| | 25 | $\{5, 7\}$ |
| 60 | 15 | $\{4, 6\}$ |
| | 40 | $\{7, 8\}$ |
| 80 | 20 | $\{3, 5\}$ |
| | 60 | $\{5, 10\}$ |
| 100 | 30 | $\{3, 6\}$ |
| | 80 | $\{5, 15\}$ |

**Table 5.3.** Results for the MIP formulation of the linear NSP on a testset of 100 larger Gaussian random matrices and larger sparsity levels.

| formulation | #opt | #nodes | time |
|---|---|---|---|
| MIP (5.5) (linearized sos1) | 12 | 4 032 423.5 | 3084.67 |
| MIP (5.10) (explicit sos1) | 16 | 2 290 380.5 | 2597.86 |
| MIP (5.6) (without sos1) | 37 | 432 792.1 | 705.02 |

stances with $n = 20$. The MIP formulations only verified the NSP for two matrices, both with $(n, m, s) = (20, 15, 3)$. The SDP relaxation shows for six more matrices that the NSP holds by producing an optimal value $< \frac{1}{2}$, namely for all matrices with $(n, m, s) = (100, 80, 5)$ and one matrix with $(n, m, s) = (80, 60, 5)$. However, for most other matrices, the MIP formulations ended up with a primal bound larger than $\frac{1}{2}$ after the time limit, which certifies that the NSP does not hold for these matrices. For seven matrices, we could not verify within the time limit whether or not the NSP holds.

As a conclusion, the formulation (5.6), which does not need sos1-constraints clearly outperforms both the formulations (5.5) and (5.10) with sos1-constraints in terms of solved instances, used nodes and solution time. If the handling of the sos1-constraints is left to SCIP, then this helps to solve more instances and speeds up the solution process, in comparison to linearizing the sos1-constraints. Thus, for testing whether a given measurement matrix satisfies the NSP for uniform recovery of sparse vectors, the formulation (5.6) should be used. Moreover, since the exact solution value of the optimization problem is not important, the computations can be safely stopped once a primal solution with solution value larger than $\frac{1}{2}$ is found, or if the dual bound gets smaller than $\frac{1}{2}$. In the latter case, the optimal solution value is guaranteed to be smaller than $\frac{1}{2}$ as well, so that the NSP is satisfied, whereas a feasible solution with solution value larger than $\frac{1}{2}$ is a certificate that the NSP is violated. By exploiting this fact, it is expected that the time needed for testing the NSP can be reduced even further, and it remains an open question to analyze the impact of this criterion for early termination.

**Nonnegative NSP** Analogously to the classical NSP, also the nonnegative NSP can be formulated as a MIP. Recall from Example (2.12.2) that the nonnegative null space property ($\mathrm{NSP}_{\geq 0}$), which characterizes uniform recovery of sparse nonnegative vectors $x \in \mathbb{R}^n_+$, reads

$$v_{\overline{S}} \leq 0 \implies \sum_{i \in S} v_i < \|v_{\overline{S}}\|_1 \quad \forall\, v \in \mathrm{null}(A)\backslash\{0\},\ \forall\, S \subseteq [n],\ |S| \leq s. \tag{$\mathrm{NSP}_{\geq 0}$}$$

This NSP is equivalent to the condition

$$\begin{aligned} \max \{ \|v_S^+\|_1 \,:\, v_{\overline{S}} \leq 0,\ Av = 0,\ \|v\|_1 \leq 1,\ v = v^+ - v^-,& \\ \mathrm{sos1}(v_i^+, v_i^-),\ i \in [n],\ |S| \leq s\} < \frac{1}{2}.& \end{aligned} \tag{5.11}$$

The constraint $v_{\overline{S}} \leq 0$ ensures that $v$ has at most $s = |S|$ nonnegative entries, and that $\|v_S^+\|_1 = \|v^+\|_1 = \sum_{i=1}^n v_i^+$. Thus, the objective function in (5.11) is already linear and does not need to be reformulated. Using the reformulation of $\mathrm{sos1}(v_i^+, v_i^-)$ in (5.4) yields the following MIP formulation of (5.11):

$$\begin{aligned} \max \quad & \sum_{i=1}^n v_i^+ \\ \text{s.t.} \quad & A(v^+ - v^-) = 0, \quad \sum_{i=1}^n (v_i^+ + v_i^-) \leq 1, \quad \sum_{i=1}^n d_i^+ \leq s, \\ & v_i^\pm \leq d_i^\pm, \quad d_i^+ + d_i^- \leq 1, \quad d_i^\pm \in \{0,1\}, \quad v_i^\pm \in [0,1] \qquad \forall i \in [n]. \end{aligned} \tag{5.12}$$

We first use the same 100 small random matrices as in the last section to evaluate the performance of the MIP formulation (5.12) for the nonnegative NSP, see (5.9) for the combinations of $(m, n, s)$. Again, we also test the MIP formulation (5.11), where the sos1-constraints are not linearized, but explicitly added as sos1-constraint in SCIP. Table 5.4 shows the number of optimally solved instances as well as the shifted geometric means of the solution times and the number of processed nodes

**Table 5.4.** Results for the MIP formulation of the nonnegative NSP on a testset of 100 Gaussian random matrices.

| formulation | # opt | #nodes | time |
|---|---:|---:|---:|
| MIP (5.12) (linearized sos1) | 100 | 2415.3 | 4.69 |
| MIP (5.11) (explicit sos1) | 92 | 18 839.6 | 22.43 |
| MIP (5.6) (linear NSP) | 100 | 27 355.8 | 36.43 |

**Table 5.5.** Results for the MIP formulation of the nonnegative NSP on a testset of 100 larger Gaussian random matrices and larger sparsity levels.

| formulation | # opt | # nodes | time |
|---|---:|---:|---:|
| MIP (5.12) (linearized sos1) | 61 | 113 724.7 | 177.86 |
| MIP (5.11) (explicit sos1) | 41 | 415 479.3 | 765.10 |
| MIP (5.6) (linear NSP) | 37 | 432 792.1 | 705.02 |

for the two formulations. Moreover, for comparison, we add the numbers of the best MIP formulation (5.6) of the linear NSP.

It turns out that this time, linearizing the sos1-constraints is clearly better than adding them explicitly as sos1-constraints. The formulation (5.12) solves all 100 instances to optimality within the time limit, whereas using explicit sos1-constraints fails to solve 8 instances within the time limit. In comparison to the linear NSP, it turns out that testing the nonnegative NSP is clearly faster, and reduces the number of used nodes and the solution time by almost one order of magnitude. The nonnegative NSP is satisfied by 14 matrices, namely the seven matrices which already satisfy the linear NSP and all matrices with $(n, m, s) = (20, 15, 3)$ as well as four of the five matrices with $(n, m, s) = (40, 15, 2)$.

When using the larger matrices with sizes depicted in Table 5.2, this again leads to instances which are much harder to solve. The formulation (5.12) with linearized sos1-constraints can only solve 61 instances within a time limit of one hour, and using explicit sos1-constraints in SCIP results in 41 optimally solved instances. The shifted geometric means of the number of nodes and the solution times are displayed in Table 5.5. As before, linearizing the sos1-constraints leads to a better performance, in contrast to the case of the linear NSP in the previous paragraph. Moreover, also the larger matrices demonstrate that testing the nonnegative NSP seems to be easier than testing the linear NSP. Overall, the nonnegative NSP could be verified for 16 matrices: all matrices with $(n, m, s) = (20, 15, 3)$, two matrices with $(n, m, s) = (20, 15, 5)$, four matrices with $(n, m, s) = (40, 25, 5)$ and all matrices with $(n, m, s) = (100, 30, 3)$. We could not make a statement about whether the

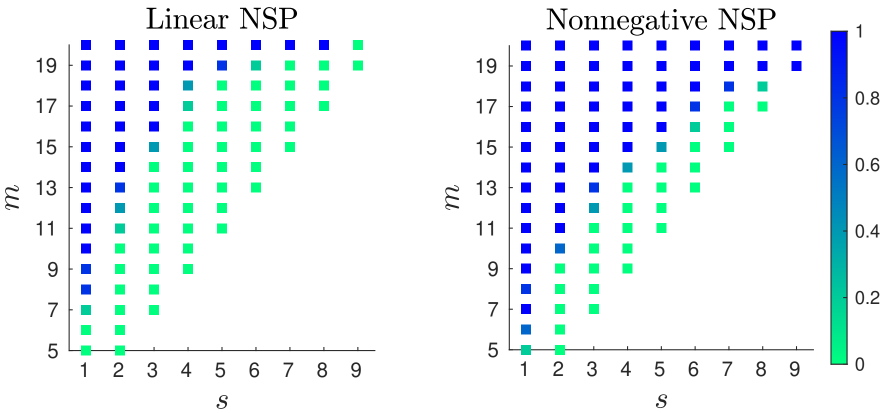


**Figure 5.1.** Empirical probability that a Gaussian random matrix satisfies the linear NSP and the nonnegative NSP for $n = 20$.

nonnegative NSP holds for all 30 matrices with

$$(n, m, s) \in \{(60, 40, 7), (60, 40, 8), (80, 60, 5), (80, 60, 10), (100, 80, 5), (100, 80, 15)\}$$

from the best primal and dual bounds obtained within the time limit. Note that among them there are 6 matrices which were shown to satisfy the linear NSP by using the SDP relaxation (5.8) without the rank1-constraint. Since the linear NSP implies the nonnegative NSP, at least these 6 matrices also satisfy the nonnegative NSP.

The results obtained for the two testsets of Gaussian random matrices again indicate that the nonnegative NSP may be satisfied for a larger combination of values $(n, m, s)$ than the linear NSP, as we have already seen in the numerical comparison in Chapter 4. In order to further support this finding, we test the linear and the nonnegative NSP for a fixed value of $n$ and all numbers of rows $m$ and sparsity levels $s$ with $5 \leq m \leq n$. Since the linear NSP cannot be satisfied for $s \geq \frac{m}{2}$, we only consider values $s < \frac{m}{2}$. In order to be able to solve most of the instances in a reasonable amount of time, we choose $n = 20$. For each combination of $(m, s)$, we created five Gaussian random matrices with $A_{ij} \sim \mathcal{N}(0, 1)$ for all $(i, j) \in [m] \times [n]$. For each combination $(m, s)$, we compute the empirical probability that an $m \times 20$ matrix satisfies the linear NSP and the nonnegative NSP of order $s$ by dividing the number of instances for which the respective NSP holds by 5 (which is the number of instances per type). The results are visualized in the heatmap in Figure 5.1. Note that the number of rows $m$ is the number of measurements that are taken. Even if the sample size of 5 matrices per size $m \times 20$ and the sparsity level $s < \frac{m}{2}$ are small, there is a clear difference between the empirical probabilities for the linear and nonnegative NSP being satisfied. The results indicate that for a fixed sparsity level $s$, fewer measurements are needed to satisfy the nonnegative NSP with high probability than to satisfy the linear NSP. Consequently, uniform recovery of every $s$-sparse nonnegative vector is guaranteed for a smaller number of measurements $m$, compared to uniform recovery of every vector, which is in line with the empirical observations in Chapter 4. Moreover, Figure 5.1 again shows a clear phase between transition violating and satisfying the NSP with high probability, which we also observed for individual recovery in Chapter 4.

In the next two sections, we consider the NSP for uniform recovery of block-sparse and block-sparse nonnegative vectors. Since those NSPs resemble the corresponding NSPs without block-structure that we discussed previously, the MIP formulations are variants of the models presented above in (5.5) and (5.12). For this reason, we drop the details and only shortly explain the derivation of the models.

**Block-Linear NSP** In the setting of Section 3.1.2, let the matrix $A \in \mathbb{R}^{m\times n}$ and the vector $x \in \mathbb{R}^n$ be block-structured, i.e.,

$$\begin{aligned} x &= \big(x[1], \dots, x[k]\big) \in \mathbb{R}^n, && v[i] \in \mathbb{R}^{n_i},\, i \in [k], \quad n_1 + \dots + n_k = n,\\ A &= \big(A[1], \dots, A[k]\big) \in \mathbb{R}^{m\times n}, && A[i] \in \mathbb{R}^{m\times n_i},\, i \in [k], \end{aligned}$$

and let $x[S]$ denote the vector where all blocks with index not in $S$ have no nonzero entries. We can assume that the null space of $A$ also consists of block-structured vectors $v \in \mathbb{R}^n$. Recall that the null space property ($\mathrm{NSP}_{q,1}$), which characterizes uniform recovery in the case of block-structured vectors, is given by

$$\|v[S]\|_{q,1} < \|v[\overline{S}]\|_{q,1}. \tag{$\mathrm{NSP}_{q,1}$}$$

Similar to the classical case without additional block-structure, this condition can be reformulated as follows:

$$\begin{aligned} &\|v[S]\|_{q,1} < \|v[\overline{S}]\|_{q,1} \quad \forall\, v \in \mathcal{N}(A)\backslash\{0\},\ \forall\, S \subseteq [k],\ |S| \le s,\\ \Leftrightarrow\ &\max\ \big\{\|v[S]\|_{q,1}\ :\ Av = 0,\ \|v\|_{q,1} \le 1,\ |S| \le s\big\} < \frac{1}{2},\\ \Leftrightarrow\ &\max\ \Big\{\sum_{i=1}^{k} y_i\, \|v[i]\|_q\ :\ Av = 0,\ \|v\|_{q,1} \le 1,\ \|y\|_\infty \le 1, \|y\|_1 \le s\Big\} < \frac{1}{2}. \end{aligned} \tag{5.13}$$

Analogously to the linear NSP, this follows from homogeneity and the definition of the dual norm. For $q = 1$, i.e., if an $\ell_1$-norm is used on the blocks $v[i]$, $i \in [k]$, a variable split on $v$ can be used to formulate the constraint $\|v\|_{q,1} \le 1$. This leads to the bilinear objective function

$$\sum_{i=1}^{k}\sum_{j=1}^{n_i} y_i\,(v[i]_j^+ + v[i]_j^-). \tag{5.14}$$

In order to remove the bilinearity and to obtain a MIP, the objective function in (5.14) can be reformulated in two different ways by introducing auxiliary variables $w[i]_j^\pm$ or $w[i]^\pm$ for all $i$, $j$ as follows:

$$w[i]_j^\pm = y_i \cdot v[i]_j^\pm \quad \Longrightarrow \quad w[i]_j^\pm \le y_i, \quad w[i]_j^\pm \le v[i]_j^\pm, \quad w[i]_j^\pm \in [0,1], \tag{5.15}$$

$$w[i]^\pm = y_i \cdot \sum_{j=1}^{n_i} v[i]_j^\pm \quad \Longrightarrow \quad w[i]^\pm \le y_i\,n_i,\, w[i]^\pm \le \sum_{j=1}^{n_i} v[i]_j^\pm,\, w[i]^\pm \in [0,n_i]. \tag{5.16}$$

Inequalities (5.15) use variables for each $i$, $j$, whereas in Inequalities (5.16), the variables are aggregated over $j$. Using the reformulation in (5.4) for the sos1-constraints on $v[i]_j^+$ and $v[i]_j^-$, which are needed due to the variable split, we obtain the following two MIP formulations:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{k}\sum_{j=1}^{n_i}\big(w[i]_j^+ + w[i]_j^-\big) \\
\text{s.t.} \quad & A(v^+ - v^-) = 0, \quad \sum_{i=1}^{k}\sum_{j=1}^{n_i}\big(v[i]_j^+ + v[i]_j^-\big) \le 1, \quad \sum_{i=1}^{k} y_i \le s, \\
& w[i]_j^\pm \le y_i, \quad w[i]_j^\pm \le v[i]_j^\pm, \quad v[i]_j^\pm \le d[i]_j^\pm \qquad \forall i \in [k],\, j \in [n_i], \\
& d[i]_j^+ + d[i]_j^- \le 1, \quad v[i]_j^\pm \in [0,1], \quad w[i]_j^\pm \in [0,1] \qquad \forall i \in [k],\, j \in [n_i], \\
& d[i]_j^\pm \in \{0,1\}, \quad y_i \in \{0,1\} \qquad \forall i \in [k],\, j \in [n_i],
\end{aligned} \tag{5.17}$$

as well as

$$\begin{aligned}
\max \quad & \sum_{i=1}^{k}\big(w[i]^+ + w[i]^-\big) \\
\text{s.t.} \quad & A(v^+ - v^-) = 0, \quad \sum_{i=1}^{k}\sum_{j=1}^{n_i}\big(v[i]_j^+ + v[i]_j^-\big) \le 1, \quad \sum_{i=1}^{k} y_i \le s, \\
& w[i]^\pm \le y_i\,n_i, \quad w[i]^\pm \le \sum_{j=1}^{n_i} v[i]_j^\pm, \quad v[i]_j^\pm \le d[i]_j^\pm \qquad \forall i \in [k],\, j \in [n_i], \\
& d[i]_j^+ + d[i]_j^- \le 1, \quad v[i]_j^\pm \in [0,1], \quad w[i]^\pm \in [0,n_i] \qquad \forall i \in [k],\, j \in [n_i], \\
& d[i]_j^\pm \in \{0,1\}, \quad y_i \in \{0,1\} \qquad \forall i \in [k],\, j \in [n_i].
\end{aligned} \tag{5.18}$$

**Block-Linear Nonnegative NSP** In the presence of an additional nonnegativity constraint on the block-structured vector, the corresponding NSP reads

$$v[\overline{S}] \leq 0 \implies \sum_{i \in S} \mathbb{1}^\top v[i] < \|v[\overline{S}]\|_{1,1} \quad \forall\, S \subseteq [k],\ |S| \leq s, \tag{NSP$_{1,1,\geq 0}$}$$

see Corollary 3.8. As in the case of the nonnegative null space property (NSP$_{\geq 0}$), the constraint $v[\overline{S}] \leq 0$ ensures that at most $s = |S|$ blocks can contain nonnegative entries. Thus, we can again use a linear objective function and write (NSP$_{1,1,\geq 0}$) as

$$\max \Big\{ \sum_{i=1}^{k} \mathbb{1}^\top v[i]^+ \,:\, A(v) = 0,\ \|v\|_{1,1} \leq 1,\ v[\overline{S}] \leq 0,\ v = v^+ - v^-,$$
$$\operatorname{sos1}(v[i]_j^+, v[i]_j^-),\ i \in [k],\ j \in [n_i],\ |S| \leq s \Big\} < \frac{1}{2}.$$

This can be modeled as the following MIP:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{k} \sum_{j=1}^{n_i} v[i]_j^+ \\
\text{s.t.} \quad & A(v^+ - v^-) = 0, \quad \sum_{i=1}^{k} \sum_{j=1}^{n_i} \big(v[i]_j^+ + v[i]_j^-\big) \leq 1, \quad \sum_{i=1}^{k} y_i \leq s, \\
& d[i]_j^+ + d[i]_j^- \leq 1, \quad v[i]_j^\pm \leq d[i]_j^\pm, \quad \sum_{j=1}^{n_i} d[i]_j^+ \leq y_i\, n_i \qquad \forall i \in [k],\ j \in [n_i], \\
& d[i]_j^\pm \in \{0,1\}, \quad v[i]_j^\pm \in [0,1] \qquad \forall i \in [k],\ j \in [n_i],
\end{aligned} \tag{5.19}$$

where we used (5.4) to model the sos1-constraints on $v[i]_j^\pm$. The constraints $\sum_{i=1}^{k} y_i \leq s$ and $\sum_{j=1}^{n_i} d[i]_j^+ \leq y_i\, n_i$ together ensure that at most $s$ blocks contain nonnegative elements, since for at most $s$ blocks, some $d[i]_j^+$ is allowed to attain the value 1, so that $v[i]_j^+$ can be nonzero.

We conducted experiments with the same matrix sizes and sparsity levels as for the linear and nonnegative NSP. The number of blocks and blocksizes are depicted in Table 5.6, where $\{3^4, 4^2\}$ means that four blocks of size three and two blocks of size four were used. Again, we generated five Gaussian random matrices $A \in \mathbb{R}^{m \times n}$ with $A_{ij} \sim \mathcal{N}(0,1)$ for each combination $(m,n,s)$ listed in Table 5.6. For each random matrix, we test the MIPs (5.17) and (5.18) in two variants: Either the linearized sos1-constraints or explicit sos1-constraints are used. The setup is exactly the same as in the previous paragraphs for the linear and the nonnegative NSP.

**Table 5.6.** Sparsity levels and sizes of the random instances used for evaluating the MIP formulations of the block-linear and block-linear nonnegative NSP.

| #cols $n$ | #rows $m$ | sparsity $s$ | #blocks $k$ | blocksizes |
|---|---|---|---|---|
| 20 | 7 | $\{2,3\}$ | 5 | 4 |
| | 15 | $\{3,5\}$ | 6 | $\{3^4,4^2\}$ |
| 40 | 15 | $\{3,4\}$ | 8 | 5 |
| | 25 | $\{5,7\}$ | 10 | 4 |
| 60 | 15 | $\{4,6\}$ | 10 | 6 |
| | 40 | $\{7,8\}$ | 20 | 3 |
| 80 | 20 | $\{3,5\}$ | 18 | $\{4^{10},5^8\}$ |
| | 60 | $\{5,10\}$ | 20 | 4 |
| 100 | 30 | $\{3,6\}$ | 21 | $\{4^{10},5^8,6,6,8\}$ |
| | 80 | 5 | 20 | 5 |
| | 80 | 15 | 25 | 4 |

Table 5.7a lists the number of solved instances within the time limit of one hour and the shifted geometric means of the number of nodes and the solution time in seconds. As can be seen, using (5.16) to linearize the objective function solves more instances within the time limit and reduces the solution time, as well as the number of used nodes. Moreover, if the sos1-constraints are explicitly added to SCIP, this improves the performance of both formulations. The number of nodes is significantly reduced and the solving time is about 20 % faster. However, the same number of instances can be solved. Interestingly, the fastest formulation (5.18) with explicit sos1-constraints can solve one instance less than the formulation (5.16) with linearized sos1-constraints.

Table 5.7b displays the results for the block-nonnegative NSP for the same 100 Gaussian random matrices. Listed are the number of solved instances within the time limit of one hour as well as the shifted geometric means of the number of nodes and the solution time in seconds. We solved (5.19) once with the linearized sos1-constraints and once using explicit sos1-constraints. Clearly, the MIP formulation of block-nonnegative NSP can be solved significantly faster than the MIP formulation of the block-linear NSP. Moreover, for the block-nonnegative NSP, using the linearized sos1-constraint is clearly faster. It also uses fewer nodes and solves more instances than using explicit sos1-constraints. Unfortunately, we could not find a single matrix satisfying either the block-linear or the block-linear nonnegative NSP.

The findings for the solution times are in line with the results in the previous paragraph. For the nonnegative NSP and the block-nonnegative NSP, using the linearized sos1-constraints is more effective, whereas for the linear NSP and the

**Table 5.7.** Results for the MIP formulation of the block-linear and the block-nonnegative NSP on a testset of 100 Gaussian random matrices.

**(a)** Block-linear NSP.

| formulation | #opt | #nodes | time |
|---|---|---|---|
| MIP (5.17) (linearized sos1) | 50 | 81 319.6 | 131.0 |
| MIP (5.18) (linearized sos1) | 55 | 68 430.4 | 115.4 |
| MIP (5.17) (explicit sos1) | 50 | 29 757.2 | 102.0 |
| MIP (5.18) (explicit sos1) | 54 | 33 537.5 | 91.0 |

**(b)** Block-linear nonnegative NSP.

| formulation | # opt | # nodes | time |
|---|---|---|---|
| MIP (5.19) (linearized sos1) | 90 | 2384.0 | 10.3 |
| MIP (5.19) (explicit sos1) | 74 | 3671.4 | 18.7 |

block-linear NSP, explicit sos1-constraints should be used to improve the performance. Furthermore, the MIP formulation of the (block-) nonnegative NSP can be solved significantly faster than the MIP formulation of the (block-) linear NSP.

It would be interesting to investigate whether there also exists an SDP reformulation of the MIP formulation for the nonnegative NSP and the block-linear (nonnegative) NSP, similar to the case of the linear NSP. An exact SDP formulation most likely would contain a rank1-constraint as well, but dropping this constraint would give a relaxation which can be used to certify the respective NSP. However, due to the explicit sign constraint $v_{\overline{S}} \leq 0$ in (NSP$_{\geq 0}$) and the block-structure in the block-linear (nonnegative) NSP, an SDP formulation seems not to be as straight-forward as in the linear case, where a simple variable change $X = xx^\top$ was sufficient.

**Low-rank Matrix NSP** In principle, the same ideas that we used above for the linear, nonnegative and block-linear (nonnegative) NSP can also be applied to obtain formulations for the null space properties characterizing uniform recovery of low-rank (positive semidefinite) and block-diagonal (positive semidefinite) matrices. However, it seems to be difficult to derive an MISDP formulation. This is due to the fact that the mentioned null space properties for matrix recovery contain conditions on the eigenvalues (or, in general, singular values) of a matrix in the null space of the measurement operator. This means, any valid formulation needs to control the eigenvalues of a matrix but must also ensure that the corresponding matrix is in the null space. For simplicity of the following considerations, we assume that all matrices are real symmetric $n \times n$ matrices.

For the case of low-rank matrix recovery without any additional side constraint, the corresponding null space property (NSP$^*$) is given as

$$\|\lambda_S(V)\|_1 < \|\lambda_{\overline{S}}(V)\|_1 \quad \forall\, V \in \big(\mathrm{null}(A) \cap \mathcal{S}^n\big) \setminus \{0\},\ \forall\, S \subseteq [n],\ |S| \leq s, \qquad (\mathrm{NSP}^*)$$

where $\lambda(V)$ denotes the vector of eigenvalues of $V$, see Example (2.12.3). Using [190, Theorem A.4], this condition is equivalent to

$$\max_{V,\, Y \in \mathcal{S}^n} \{\langle Y, V\rangle_{\mathrm{F}} \,:\, A(V) = 0,\ \|V\|_* \leq 1,\ \|Y\|_2 \leq 1, \|Y\|_* \leq s\} < \frac{1}{2}, \tag{5.20}$$

where $\|Y\|_2 \coloneqq \max\{|\lambda_i(Y)|\}$ denotes the operator norm of $Y$, i.e., the largest eigenvalue in absolute value. Using the Schur complement, the operator norm $\|X\|_2$ can be written as follows:

$$\|X\|_2 \leq s \quad \Leftrightarrow \quad s^2\,\mathbb{I} - XX^\top \succeq 0 \quad \Leftrightarrow \quad \begin{pmatrix} s\mathbb{I} & X \\ X^\top & s\mathbb{I} \end{pmatrix} \succeq 0. \tag{5.21}$$

Recht et al. [210, Proposition 2.1] prove that the nuclear and the operator norm are dual to each other. The proof uses SDP duality and also shows that the nuclear norm can be computed using one of the following SDPs:

$$\begin{aligned} \|X\|_* = \max\, &\langle X, Y\rangle_{\mathrm{F}} & = \max\quad &\langle X, Y\rangle_{\mathrm{F}} & = \min\quad &\tfrac{1}{2}\big(\mathrm{tr}(W_1) + \mathrm{tr}(W_2)\big) \\ \text{s.t. } &\begin{pmatrix} \mathbb{I} & Y \\ Y^\top & \mathbb{I} \end{pmatrix} \succeq 0 & \text{s.t.}\quad &\|Y\|_2 \leq 1 & \text{s.t.}\quad &\begin{pmatrix} W_1 & X \\ X^\top & W_2 \end{pmatrix} \succeq 0. \end{aligned}$$

Thus, the constraint $\|V\|_* \leq 1$ is equivalent to the existence of $W_1$, $W_2 \in \mathcal{S}^n$ with $\mathrm{tr}(W_1) + \mathrm{tr}(W_2) \leq 2$ and

$$\begin{pmatrix} W_1 & X \\ X^\top & W_2 \end{pmatrix} \succeq 0.$$

The constraint $\|Y\|_* \leq s$ can be reformulated analogously. By (5.21), the remaining constraint $\|Y\|_2 \leq 1$ is equivalent to

$$\begin{pmatrix} \mathbb{I} & X \\ X^\top & \mathbb{I} \end{pmatrix} \succeq 0.$$

This shows that (5.20) is a bilinear SDP formulation for the matrix null space property (NSP$^*$), which is analogous to the bilinear formulation (5.2) for the NSP for sparse vectors above. There, we used sos1-constraints for the entries of a vector to overcome the bilinearity. However, there is no direct analog for matrices, since there is no natural ordering for the eigenvalues of a matrix, as opposed to the entries of a vector. Thus, there is no easy way to split a matrix $V$ into two matrices $V^+$ and $V^-$ with only positive and negative eigenvalues, respectively, and to ensure that all eigenvalues of $V^+$ and $V^-$ are eigenvalues of the original matrix $V$ as well. Of course, a simple "brute-force" approach to deal with this problem is to explicitly use the eigenvalue decomposition $V = UDU^\top$, where $U$ is a orthogonal matrix and $D$ is a diagonal matrix containing the eigenvalues. This yields the following "trilinear" problem to compute the null space property (NSP$^*$):

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} \big(y_i\, \lambda_i^+ + y_i\, \lambda_i^-\big) \\
\text{s.t.} \quad & A(UDU^\top) = 0, \qquad \sum_{i=1}^{n} y_i \le s, \qquad \sum_{i=1}^{n} \big(\lambda_i^+ + \lambda_i^-\big) \le 1, \\
& \operatorname{sos1}(\lambda_i^+, \lambda_i^-), \qquad \lambda_i^\pm \in [0,1], \quad y_i \in \{0,1\} \qquad \forall\, i \in [n], \\
& U \text{ unitary}, \quad D = \operatorname{Diag}(\lambda_1, \dots, \lambda_n).
\end{aligned}$$

This is a mixed-integer nonlinear problem (MINLP), which has the matrix entries of $U$ as well as the eigenvalues of $V$ as variables. Since $V = UDU^\top$, the entries of $V$ appear implicitly as variables. For the recovery of low-rank positive semidefinite matrices and for the recovery of block-diagonal (positive semidefinite) matrices, the same problem emerges, so that it remains an open question to find an MISDP formulation without bilinearities for these NSPs.

In the next section, we consider the restricted isometry property, which is another well-known condition that guarantees uniform recovery of sparse vectors by $\ell_1$-minimization.

## 5.2 A Mixed-Integer Semidefinite Programming Formulation for the Restricted Isometry Property

As outlined in Chapter 1, null space properties are not the only conditions which guarantee uniform recovery of sparse vectors using $\ell_1$-minimization. Another well-known example of such a condition is the restricted isometry property (RIP). Historically, the RIP was developed prior to the NSP as a condition for uniform recovery of sparse vectors. In contrast to the NSP, it is only a sufficient condition for uniform recovery. In this section, we turn our attention towards the RIP, and consider an approach to test this condition for a given measurement matrix $A$. It is known that this problem can be formulated as an MISDP. In the following, we shortly introduce the RIP and its implications for the recovery of sparse vectors. Afterwards, we present an MISDP formulation of the RIP, and consider special components that can be used in the solution process. Numerical experiments for the MISDP formulation of the RIP and the special components are postponed to the next chapter. There, we introduce several presolving techniques for general MISDPs and evaluate their impact numerically on various classes of MISDPs with a special focus on the MISDP formulation of the RIP.

The RIP has been introduced by Candès and Tao in [38, 45], who showed in [38] that it can be used to guarantee exact uniform recovery of sparse vectors by $\ell_1$-minimization. The extension of this result to stable and robust uniform recovery is due to Candès et al. [39]. An MISDP formulation for computing the RIP has been obtained by Gally and Pfetsch [111], which builds upon an asymmetric version of the RIP introduced by Foucart and Lai [103]. Computational results for the MISDP formulation of the RIP can be found in [111], [145] and in Section 6.6, which is taken from [174].

**Definition 5.2.** *Let $s \geq 0$ be a nonnegative integer. A matrix $A \in \mathbb{R}^{m \times n}$ satisfies the* restricted isometry property (RIP) *of order $s$ with constant $\delta \geq 0$, if*

$$(1-\delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\|x\|_2^2 \tag{5.22}$$

*holds for all $s$-sparse $x \in \mathbb{R}^n$, i.e., for all $x \in \Sigma_s := \{x \in \mathbb{R}^n : \|x\|_0 \leq s\}$. The smallest constant $\delta$ such that* (5.22) *holds for all $x \in \Sigma_s$ is called the $s$-th restricted isometry constant (RIC) $\delta_s$.*

Candès [41] obtained the well-known sufficient condition $\delta_{2s} < \sqrt{2} - 1$ for stable and robust uniform recovery of $s$-sparse vectors using $\ell_1$-minimization. Since then, this condition has been refined, improved and modified numerous times, see, e.g., the notes at the end of Chapter 6 in [104] for further references. Cai and Zhang [37] obtained the sharp recovery guarantee $\delta_{2s} < 1/\sqrt{2}$ for sparse vectors and also low-rank matrices.

It is known that the problem of computing the RIC $\delta_s$ is $\mathcal{NP}$-hard, see Tillmann and Pfetsch [237]. In order to formulate an MISDP to compute the RIC $\delta_s$ for a given matrix $A \in \mathbb{R}^{m \times n}$ and a given sparsity level $s$, it is useful to split the lower and upper bound in (5.22) into two separate conditions. This leads to the following lower and upper restricted isometry constants, which were proposed by Foucart and Lai [103].

**Definition 5.3.** *Let $A \in \mathbb{R}^{m\times n}$ and $s \geq 0$ be a nonnegative integer. The* lower *and* upper restricted isometry constants *$\alpha_s$ and $\beta_s$ order $s$ of $A$ are defined as*

$$\alpha_s := \max\{\alpha \geq 0 \,:\, \alpha^2\|x\|_2^2 \leq \|Ax\|_2^2 \;\forall\, x \in \Sigma_s\}, \tag{5.23}$$

$$\beta_s := \min\{\beta \geq 0 \,:\, \beta^2\|x\|_2^2 \geq \|Ax\|_2^2 \;\forall\, x \in \Sigma_s\}, \tag{5.24}$$

*respectively. The quotient $\gamma_s := \beta_s^2/\alpha_s^2$ for $\alpha_s \neq 0$ is called the* restricted isometry ratio (RIR)*.*

The lower/upper RIC from Definition 5.3 is a generalization of the RIC from Definition 5.2, since a RIC of $\delta_s$ implies that the RIR is at most $(1+\delta_s)/(1-\delta_s)$, and a lower/upper RIC $\alpha_s$ and $\beta_s$, respectively, implies a RIC $\delta_s = \max\{1-\alpha_s^2, \beta_s^2-1\}$. Foucart and Lai [103] show the corresponding sufficient condition $\gamma_{2s} \leq 4\sqrt{2}-3$ for uniform recovery.

By scaling the vector $x \in \Sigma_s$ to be of unit norm, the (squared) lower and upper RIC in (5.23) and (5.24) can be equivalently written as

$$\alpha_s^2 = \min\{\|Ax\|_2^2 \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq s\}, \tag{5.25}$$

$$\beta_s^2 = \max\{\|Ax\|_2^2 \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq s\}. \tag{5.26}$$

Let $x^\star$ be an optimal solution of either of the two problems and $S = \operatorname{supp}(x^\star)$ be its support with $k := \|x^\star\|_0$. Consider the submatrix $A_S \in \mathbb{R}^{m\times k}$ indexed by columns in $S$. Then $\tilde{A} = A_S^\top A_S \in \mathbb{R}^{k\times k}$ is symmetric positive semidefinite. By the Rayleigh-Ritz theorem (see, e.g., , Horn and Johnson [130, Thm. 4.2.2]) we have

$$\begin{aligned}\max_{y\in\mathbb{R}^k}\{\|\tilde{A}y\|_2^2 \,:\, \|y\|_2^2 = 1\} &= \lambda_{\max}(\tilde{A})^2,\\ \min_{y\in\mathbb{R}^k}\{\|\tilde{A}y\|_2^2 \,:\, \|y\|_2^2 = 1\} &= \lambda_{\min}(\tilde{A})^2,\end{aligned} \tag{5.27}$$

i.e., computing the lower and upper RIC as defined in (5.23) and (5.24) are sparse eigenvalue problems, which are of interest in their own regard. Moreover, the problem (5.26) is also known as *sparse principal component analysis (SPCA)*, which has been widely studied in the literature as well. Since this is not the topic of this thesis, we only refer to Zou et al. [258], where SPCA was introduced, and to Bertsimas et al. [24] for a problem-specific approach to solve SPCA at scale. Note that [174, Appendix A.4] also shows that the lower and upper RICs as defined in (5.25) and (5.26) are in fact sparse eigenvalue problems.

**An MISDP Formulation for the RIP** Recall that $X \succeq 0$ denotes that the matrix $X$ is symmetric and positive semidefinite. Moreover, $\mathcal{S}^n$ denotes the set of symmetric $n \times n$ matrices. Consider the semidefinite lifting $X = xx^\top$, which is equivalent to $X \succeq 0$ and $\operatorname{rank}(X) = 1$. By using this lifting, the formulation of the lower and upper RIC $\alpha_s^2$ and $\beta_s^2$ in (5.25) and (5.26) can be written as the following rank-1 constrained SDP:

$$\begin{aligned} \max/\min \quad & \langle A^\top A, X\rangle_{\mathrm{F}} \\ \text{s.t.} \quad & \operatorname{tr}(X) = 1, \quad \|\operatorname{vec}(X)\|_0 \leq s^2, \quad X \succeq 0, \quad \operatorname{rank}(X) = 1, \end{aligned}$$

where $\operatorname{vec}(X)$ denotes the vectorization of the matrix $X$, that is, the vector obtained by concatenating all columns of $X$ into a vector. The rank-constraint implies that every optimal solution $X^*$ satisfies $X^* = x^*(x^*)^\top$ for a vector $x^* \in \mathbb{R}^n$. The $\ell_0$-constraint $\|\operatorname{vec}(X)\|_0 \leq s^2$ can be modeled by introducing binary variables $z_i$. Then, we arrive at the following formulation, which is due to Gally and Pfetsch [111]:

$$\begin{aligned} \max/\min \quad & \langle A^\top A, X\rangle_{\mathrm{F}} && (5.28a) \\ \text{s.t.} \quad & \operatorname{tr}(X) = 1, && (5.28b) \\ & -z_j \leq X_{ij} \leq z_j \quad \text{for } i,\ j \in [n], && (5.28c) \\ & \sum_{i=1}^{n} z_i \leq k, && (5.28d) \\ & X \succeq 0, && (5.28e) \\ & \operatorname{rank}(X) = 1, && (5.28f) \\ & z \in \{0,1\}^n. && (5.28g) \end{aligned}$$

This is an MISDP formulation for computing the upper and lower RIC $\alpha_s^2$ and $\beta_s^2$. The additional rank-constraint on $X$ can be dropped without losing exactness of the formulation, since in [111] (and Li and Xie [157] as well as Bertsimas et al. [24]) it is proved that there exists an optimal rank-1 solution $X^*$. Thus, $X^* = x^*(x^*)^\top$ for some $x^* \in \mathbb{R}^n$ with $\|x^*\|_0 \leq k$. Let $S = \operatorname{supp}(x^*)$. Then $x_S^*$ is an eigenvector for a maximal or minimal eigenvalue of $A_S^\top A_S$, depending on the objective sense, see (5.27). Moreover, in [111] it is also shown that the bounds in (5.28c) can be strengthened to

$$-\tfrac{1}{2} z_j \leq X_{ij} \leq \tfrac{1}{2} z_j \qquad (5.29)$$

for all $i \neq j \in [n]$. However, the bounds on the diagonal entries $X_{ii}$ cannot be tightened.

For a discussion of MISDPs, we refer to the subsequent Chapter 6. There, several presolving routines for general MISDPs are introduced and solution approaches for general MISDPs are reviewed. The presolving methods are numerically tested on five different classes of MISDPs, among them the MISDP (5.28) for testing the RIP. In the next section, we instead present special methods that can applied or exploited when solving the MISDP (5.28) for testing the RIP. A numerical evaluation of the presented methods also follows in Chapter 6 together with the analysis of the introduced presolving methods.

## 5.3 Special Components for the MISDP Formulation of the RIP

Now we consider special components for solving the MISDP formulation (5.28) of the lower and upper RIC $\alpha_s^2$ and $\beta_s^2$. The content of this section is based on unpublished joint work with Marc E. Pfetsch.

**Complete Description of the Feasible Set** One interesting question is whether one can describe the convex hull of the feasible set. For this define

$$S := \Big\{(X, z) \in \mathcal{S}_+^n \times [0,1]^n \,:\, \operatorname{tr}(X) = 1,\ \sum_{i=1}^{n} z_i \leq k,\ 0 \leq X_{ii} \leq z_i \ \forall\, i \in [n]\Big\},$$

which is the relaxation of (5.28). Note that since $X_{ii} = 0$ implies $X_{ij} = 0$ for all $j \in [n]$, the constraints $-z_j \leq X_{ij} \leq z_j$ are implied by $0 \leq X_{ii} \leq z_i$, so that they can in principle be omitted. The integer hull of $S$ is

$$S_I := \operatorname{conv}\Big\{(X, z) \in \mathcal{S}_+^n \times \{0,1\}^n \,:\, \operatorname{tr}(X) = 1,\ \sum_{i=1}^{n} z_i \leq k,\ 0 \leq X_{ii} \leq z_i \ \forall\, i \in [n]\Big\}.$$

In general, we have $S \neq S_I$: If $k = 1$, then all off-diagonals have to be zero in $S_I$, but not necessarily in $S$. In order to prove $S \neq S_I$ in the general case $k \geq 2$ the valid inequality in the next lemma will be used. We define the set

$$C := \Big\{(X, z) \in \mathcal{S}_+^n \times \{0,1\}^n \,:\, \operatorname{tr}(X) = 1,\ \sum_{i=1}^{n} z_i \leq k,\ 0 \leq X_{ii} \leq z_i \ \forall\, i \in [n]\Big\}. \tag{5.30}$$

Furthermore, we use the short notation $\sum_{i \neq j}$ to denote a sum which ranges over all $(i, j) \in [n] \times [n]$ with $i \neq j$.

**Lemma 5.4.** *Let* $(X, z) \in C$*, where* $C$ *is defined as in* (5.30)*. Then*

$$-k + 1 \leq \sum_{i \neq j} X_{ij} \leq k - 1. \tag{5.31}$$

*Proof.* For $i$, $j \in [n]$, let $v = e_i - e_j \in \mathbb{R}^n$. Then, because $X \succeq 0$, we get

$$0 \leq v^\top X v = X_{ii} + X_{jj} - 2\, X_{ij} \quad \Leftrightarrow \quad 2\, X_{ij} \leq X_{ii} + X_{jj}. \tag{5.32}$$

Define $I = \{(i, j) \in [n] \times [n] \; : \; i \neq j,\, X_{ii} \neq 0,\, X_{jj} \neq 0\}$. Then, summing all off-diagonal positions $(i, j) \in [n] \times [n]$ with $i \neq j$ yields

$$\begin{aligned}\sum_{i \neq j} X_{ij} = \sum_{(i,j) \in I} X_{ij} &\leq \sum_{(i,j) \in I} \tfrac{1}{2}(X_{ii} + X_{jj}) = \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j:(i,j) \in I} \bigl(X_{ii} + X_{jj}\bigr) \\ &= \sum_{i=1}^{n} \sum_{j:(i,j) \in I} X_{ii} \leq (k-1) \sum_{i=1}^{n} X_{ii} = k - 1,\end{aligned}$$

where the first inequality is due to (5.32). The third equality follows, since $(i, j) \in I$ implies $(j, i) \in I$ as well, and the last inequality uses that at most $k$ diagonal entries of $X$ are nonzero. Finally, the last equality is due to $\operatorname{tr}(X) = 1$. Using $X \succeq 0$ and $v = e_i + e_j \in \mathbb{R}^n$ yields

$$0 \leq v^\top X v = X_{ii} + X_{jj} + 2\, X_{ij} \quad \Leftrightarrow \quad 2\, X_{ij} \geq -\bigl(X_{ii} + X_{jj}\bigr).$$

Then, as above,

$$\sum_{i \neq j} X_{ij} = \sum_{(i,j) \in I} X_{ij} \geq \sum_{(i,j) \in I} -\tfrac{1}{2}\bigl(X_{ii} + X_{jj}\bigr) \geq= -k + 1. \qquad \square$$

Using Lemma 5.4, we can show $S \neq S_I$ for $k \geq 2$.

**Lemma 5.5.** *Let* $2 \leq k \leq n - 1$ *and*

$$\hat{z} := \bigl[\,\underbrace{\tfrac{k}{k+1}, \dots, \tfrac{k}{k+1}}_{k+1}, 0, \dots, 0\bigr]^\top, \qquad \hat{X} := \begin{pmatrix} \frac{1}{k+1} \mathbf{1}_{k+1} & \mathbf{0}_{n-k-1} \\ \mathbf{0}_{k+1} & \mathbf{0}_{n-k-1} \end{pmatrix},$$

*where* $\mathbf{1}_k$ *and* $\mathbf{0}_k$ *are all-one resp. all-zero matrices of dimension* $k \times k$*. Then* $(\hat{X}, \hat{z}) \in S$*, but* $(\hat{X}, \hat{z}) \notin S_I$*.*

*Proof.* By definition, we have $(\hat{X}, \hat{z}) \in S_I$, if there exists a convex combination in terms of $(X, z) \in C$. If $\hat{X}$ is written as convex combination of matrices $X \in C$, then $\hat{X}$ also needs to satisfy the bounds in (5.31) by Lemma 5.4. However, $\sum_{i \neq j} \hat{X}_{ij} = k$, so that $\hat{X} \notin S_I$. □

Note that after adding the inequality $\sum_{i \neq j} X_{ij} \leq k - 1$ to (5.28), Corollary 2.32 in [110] shows that there still exists an optimal rank-1 solution $X^*$ of (5.28). However, if we add another valid inequality then we might lose this property. Consequently, dropping the rank-constraint would no longer yield an exact RIP formulation, but only a relaxation.

**Nonnegativity by the Perron-Frobenius Theorem** If the matrix $A$ is componentwise nonnegative, we can apply the Perron-Frobenius theorem.

**Theorem 5.6** (Perron-Frobenius Theorem, see Gantmacher [115, Chapter 2, Thm. 3]). *A nonnegative matrix $A$ has a nonnegative maximal eigenvalue $\lambda \geq 0$. The corresponding eigenvectors have nonnegative entries.*

Recall that the problem of computing the lower and upper RIC $\alpha_s^2$ and $\beta_s^2$ are sparse eigenvalue problems, see (5.25) and (5.26). Thus, Theorem 5.6 implies the following.

**Lemma 5.7.** *Let $A \geq 0$ componentwise. Then, there exists an optimal solution $x^*$ of* $\max\{\|Ax\|_2^2 \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq s\}$ *which has nonnegative entries.*

*Proof.* Let $x^*$ be an optimal solution of $\max\{\|Ax\|_2^2 \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq s\}$ with $S = \operatorname{supp}(x^*)$. By (5.27), $x^*_S$ is an eigenvector for a maximal eigenvalue of the symmetric positive semidefinite matrix $A_S^\top A_S$. Either $x^*_S \geq 0$ componentwise, or there exists another maximal eigenvalue with a corresponding nonnegative eigenvector $\tilde{x}_S$ by Theorem 5.6. Setting $\tilde{x}_S$ to zero outside of $S$ yields the vector $\tilde{x}$, which is an optimal solution of $\max\{\|Ax\|_2^2 \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq s\}$ as well. □

This implies that we can restrict $X \geq 0$ when computing the upper RIC $\beta_s^2$ and write

$$0 \leq X_{ij} \leq z_j \quad \text{for } i,\ j \in [n]$$

instead of (5.28c) if $A \geq 0$ componentwise. Since Lemma 5.7 only holds for the maximization problem (5.26), but not the minimization problem (5.25), adding the constraint $X_{ij} \geq 0$ if $A \geq 0$ is not feasible for computing the lower RIC $\alpha_s^2$.

**Sparsification of Eigenvector Cuts** In general, MISDPs can be solved by branch-and-bound algorithms, where in each node, either an SDP or an LP relaxation is solved, see Section 6.1 for a brief description. If the LP relaxation is used, then the positive semidefiniteness constraint needs to be ensured by adding linear cuts. Such cuts, which are called *eigenvector cuts*, can be generated by using eigenvectors to negative eigenvalues of the current relaxation solution. If $X^*$ is a relaxation solution which is not positive semidefinite, let $v^*$ be an eigenvector to a negative eigenvalue of $X^*$. Then,

$$(v^*)^\top X v^* \geq 0$$

is a valid linear inequality, which cuts $X^*$ off, see Section 6.1 for more details. These inequalities are typically dense, i.e., they contain many nonzero coefficients. It is known that adding dense cuts can lead to stronger relaxations, but also increases the time needed to solve them, whereas sparse cuts can result in significant performance increases, see, e.g., Dey and Molinaro [63]. Moreover, Blekherman et al. [27] show that the cone of positive semidefinite matrices can be well-approximated by $k \times k$ minors, which motivates to employ $k$-sparse eigenvector cuts. Such sparse eigenvector cuts for the MISDP (5.28) can be obtained by exploiting that the original formulation (5.24) and (5.25) are sparse eigenvalue problems, as already done in Lemma 5.7 in order to derive nonnegativity of $X$ if $A \geq 0$ componentwise. Sparsifying eigenvector cuts has been considered by Qualizza et al. [207], and computing multiple sparse eigenvector cuts directly has been proposed by Dey et al. [64]. Both papers mainly considered SDP relaxations of quadratic problems, but the technique proposed in [64] can also be applied for the MISDP formulation (5.28) for the RIP to generate sparse eigenvector cuts.

The main idea is to repeatedly compute a single sparse eigenvector cut with support set $I$. This cut is not added, but used to obtain eigenvector cuts for the submatrix restricted to the support set $I$. These cuts are automatically as sparse as the original sparse cut. However, the authors in [64] empirically observe that using only one support set has no significant effect on the performance. Thus, they propose an iterative procedure that varies the considered support set. This is achieved by subtracting the rank-1 matrix formed by the computed sparse eigenvector and the corresponding unit norm eigenvalue from the current matrix. Afterwards, a sparse eigenvector cut for the updated matrix is computed, which possibly leads to a new support set $J$. This procedure is shown to terminate in a finite number of iterations, see [64, Lemma 3]. In order to solve the sparse eigenvalue problem, the Truncated Power Method (TPower) by Yuan and Zhang [255] can be used. This method is an efficient heuristic with theoretical guarantees for computing the largest sparse eigenvalue, that is, the largest eigenvalue so that the corresponding eigenvector is sparse.

For the RIP problem, it is meaningful to use the sparsity level of the lower and upper RIC as desired sparsity for the eigenvector cuts. In the numerical experiments in the next chapter, we will see that this significantly speeds up the solution time for the LP-based branch-and-bound approach.

**Minimization Variant** In the following, we show that the problems of computing the lower and upper RIC are equivalent in the sense that the minimization problem in (5.25) for the lower RIC can be transformed into an instance of the maximization problem (5.26) for the upper RIC.

**Lemma 5.8.** *The problem* (5.25) *for computing the lower RIC can be transformed to an instance of the problem* (5.26) *for computing the upper RIC.*

*Proof.* Define $\lambda \coloneqq \lambda_{\max}(A^\top A)$. Since $A^\top A$ is positive semidefinite, we have $\lambda \geq 0$. Then we can rewrite (5.25) as

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \{x^\top A^\top A x \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq x\} \\ =& \min_{x \in \mathbb{R}^n} \{x^\top (A^\top A - \lambda\, I) x + \lambda x^\top x \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq k\} \\ =& \min_{x \in \mathbb{R}^n} \{x^\top (A^\top A - \lambda\, I) x \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq k\} + \lambda \\ =& \lambda - \max_{x \in \mathbb{R}^n} \{x^\top (\lambda\, I - A^\top A) x \,:\, \|x\|_2^2 = 1,\ \|x\|_0 \leq k\}. \end{aligned}$$

The matrix $\lambda\, I - A^\top A$ is positive semidefinite, since

$$v^\top (\lambda\, I - A^\top A) v = \lambda v^\top v - \underbrace{v^\top A^\top A v}_{\leq \lambda\, v^\top v} \geq 0$$

for all $v \in \mathbb{R}^n$. Thus, there exists $\hat{A}$ with $\hat{A}^\top \hat{A} = \lambda\, I - A^\top A$. Then solving Problem (5.26) with $\hat{A}$ instead of $A$ yields the value of (5.25). □

As a consequence of this theorem, also the problem of computing the lower RIC is equivalent to the sparse PCA problem. However, the matrix $\lambda\, I - A^\top A$ is usually not nonnegative, so that Theorem 5.6 cannot be applied.

A numerical evaluation of the presented methods will follow at the end of the next chapter, which treats presolving techniques for general MISDPs. The impact of the introduced presolving methods is tested on different MISDPs, among them the RIP-formulation (5.28), so that it is meaningful to evaluate the special components for the RIP introduced in this section also in combination with the general presolving techniques. This evaluation is conducted in in Section 6.6.4.

CHAPTER 6

# Presolving for Mixed-Integer Semidefinite Optimization

In the last chapter, we have introduced a mixed-integer semidefinite program to test whether a given measurement matrix satisfies the RIP and thus allows for uniform recovery of sufficiently sparse vectors $x \in \mathbb{R}^n$. Moreover, we discussed several properties of this MISDP formulation. In general, different approaches to solve MISDPs based on branch-and-bound methods are known, which are shortly introduced in Section 6.1. However, few additional techniques that can be exploited throughout the solution process have been investigated in the literature. This motivates to search for presolving or propagation routines, similar to the MIP case, where such techniques are widely applied with overwhelming success. Thus, we introduce several methods that can be used for presolving for and propagation in general MISDPs in this chapter. All proposed methods are implemented in the general MISDP solver SCIP-SDP [220] and we will evaluate these methods numerically. The content of this chapter is taken from the preprint [174], which is accepted for publication in an international journal, and is joint work with Marc E. Pfetsch.

Presolving is one of the cornerstones of generic mathematical optimization solvers. It changes an instance into an equivalent one that is hopefully easier to solve. This can often be achieved by removing variables or constraints as well as tightening coefficients or bounds of variables. As in the literature, we use the terms presolving and preprocessing interchangeably.

In SCIP [219], which employs a branch-and-bound approach, presolving is applied in rounds before the solution process starts. In each round of this so-called *root node presolving*, various methods are tried. These methods include general techniques and also techniques for specific types of constraints such as linear constraints. Presolving ends if a limit of rounds is reached or if a round did not find any further deductions such as tighter bounds or removed variables or constraints. Using more than one round for presolving implies that the methods influence each other. If a constraint is added, the next round may take this new constraint into account and find further reductions. Consequently, using several presolving techniques may lead to simplifications of the problem which could not have been deduced by one of the techniques alone. Moreover, presolving not only happens before the solution process starts, but also within the branch-and-bound tree. Whenever a node is finished and the algorithm moves to a new node, so-called *node presolving* is applied for the relaxation within this node. Since this relaxation typically differs slightly from the relaxation of its parent nodes, it is meaningful to apply presolving. For all constraints which can be handled by SCIP, such as linear constraints, the same presolving as in SCIP is automatically applied by SCIP-SDP as well, since SCIP-SDP builds upon SCIP. However, this does not include presolving for SDP constraints. In this chapter, we introduce several new presolving techniques which specifically take the SDP constraint into account.

If the underlying solution process can in principle result in an exponential runtime behavior, such presolving can have an impressive impact. For instance, Bixby and Rothberg [26] report a slowdown factor of 10.8 when solving MIPs with disabled root node presolving for CPLEX 8.0; this factor was confirmed by Achterberg and Wunderling [3] for CPLEX 12.5. For mixed-integer nonlinear programs (MINLPs), Puranik and Sahinidis [205] demonstrate the importance of presolving and bound tightening: using presolving significantly speeds up the solution process and increases the number of solved instances within the time limit for the solvers BARON, COUENNE, and SCIP. It turns out that bound tightening is essential for strengthening relaxations of nonconvex problems. Note that the instances in all of these publications come from publically available benchmark libraries and are quite diverse and generic. Indeed, presolving is very useful for instances that have been generated by modeling languages. The impact of presolving of course depends on the particular instances and might be less effective for instances that come from a less generic source or are tuned ("presolved") by humans.

In this chapter, we consider general MISDPs of the form:

$$\begin{aligned} \inf\quad & b^\top y \\ \text{s.t.}\quad & \sum_{k=1}^{m} A^k\, y_k - A^0 \succeq 0, \\ & \ell_i \le y_i \le u_i && \forall\, i \in [m], \\ & y_i \in \mathbb{Z} && \forall\, i \in I, \end{aligned} \tag{6.1}$$

with symmetric matrices $A^k \in \mathcal{S}^n$ for $k \in [m]_0 := \{0,\dots,m\}$, $b \in \mathbb{R}^m$, and bounds $\ell_i \in \mathbb{R} \cup \{-\infty\}$ as well as $u_i \in \mathbb{R} \cup \{\infty\}$ for all $i \in [m] := \{1,\dots,m\}$. The set of indices of integer variables is given by $I \subseteq [m]$. Recall that for a symmetric matrix $M \in \mathcal{S}^n$, the notation $M \succeq 0$ indicates that $M$ is positive semidefinite. We use the notation $A(y) := \sum_{k=1}^m A^k\, y_k - A^0$ for $y \in \mathbb{R}^m$ throughout this chapter. Note that in some applications, e.g., reformulations of combinatorial optimization problems, it is more natural to have a positive semidefinite matrix variable $X \succeq 0$, which leads to an equivalent "primal" version of (6.1). In the following remark we outline the equivalence and also explain how to reformulate an MISDP in one form into the other. Our presentation and implementation in the solver SCIP-SDP (see below in Section 6.1), however, is based on the form in (6.1).

**Remark 6.1.** Apart from the so-called "dual" form (6.1) of an MISDP, one can also consider the corresponding "primal" form:

$$\begin{aligned} \sup \quad & \langle A^0, X\rangle_{\mathrm{F}} \\ \text{s.t.} \quad & \langle A^i, X\rangle_{\mathrm{F}} = b_i && \forall\, i \in [m], \\ & L_{ij} \le X_{ij} \le U_{ij} && \forall\, i,\, j \in [n], \\ & X_{ij} \in \mathbb{Z} && \forall\, (i,\, j) \in I \times I, \\ & X \succeq 0, \end{aligned} \tag{6.2}$$

where $\langle A, B\rangle_{\mathrm{F}}$ is the Frobenius inner product defined in (1.1). The bounds are given by $L_{ij} \in \mathbb{R} \cup \{-\infty\}$, $U_{ij} \in \mathbb{R} \cup \{\infty\}$ for all $i,\, j \in [m]$.

We note that (6.1) and (6.2) are equivalent: Indeed, starting from (6.1), one can define $Z = \sum_{i=1}^n A^i\, y_i - A^0$. The "primal" variables are

$$X = \begin{pmatrix} Z & 0 \\ 0 & \mathrm{Diag}(y) \end{pmatrix} \in \mathbb{R}^{(n+m)\times(n+m)},$$

where $\mathrm{Diag}(y)$ denotes a diagonal matrix containing $y$ on the diagonal (possibly $y$ has to be split into two nonnegative variables). The $n^2$ equations $Z = \sum_{i=1}^n A^i\, y_i - A^0$ can then be written in the form $\langle B^i, X\rangle_{\mathrm{F}} = d_i$ for appropriate matrices $B^i$ and scalars $d_i$, $i \in [n^2]$.

Conversely, given (6.2), using the Gauss algorithm on the equations $\langle A^i, X\rangle_{\mathrm{F}} = b_i$, one can express the $n^2$ variables in $X$ by introducing $r := n^2 - m$ variables $y$ as $X = \sum_{i=1}^r B^i\, y_i - B^0$ with appropriate matrices $B^i$, $i \in [r]_0$. In both directions, the objective and variable bounds can be chosen appropriately. These transformations often simplify for particular problems. Besides, the relaxations of (6.1) and (6.2) are dual to each other, i.e., the relaxation of (6.1) provides an upper bound for the relaxation of (6.2).

While for specific types of MISDPs, several presolving methods are known, this chapter focuses on presolving for generic MISDPs. We introduce several new techniques and provide a computational evaluation of different variants. Often these methods can be seen as a generalization of presolving for mixed-integer programs. We note that several methods that we describe can be performed in node presolving as well. In particular, this includes propagation of variable bounds, i.e., tightening of some variable bounds based on the bounds of other variables.

This chapter is structured as follows. Section 6.1 starts with a description of solution approaches for (6.1) and reviews the literature on presolving techniques for MIPs. Furthermore, known and easy presolving methods for MISDPs are mentioned. We then present several valid linear inequalities in Section 6.2. These can be added during presolving and are then used for further presolving steps. In Section 6.3, we turn our attention to presolving based on $2 \times 2$ minors of positive semidefinite matrices $A(y)$. This involves variable bounds derived separately from upper bounds on diagonal and off-diagonal entries. As a next step, we present a method to tighten variable bounds in Section 6.4. We prove that iteratively applying this bound tightening converges to a best bound, which can also be computed by solving a single SDP (Section 6.4.1). Such a best bound is the tightest bound which holds for any solution of the MISDP. Instead of solving the possibly large SDP to compute a best bound we furthermore show that each single bound tightening application corresponds to an SDP with one variable, which can be solved using a semismooth Newton method, see Section 6.4.2. With similar techniques, one can also compute the tightest scaling of the constraint matrices $A^k$ that does not change the feasible region; this generalizes coefficient tightening, see Section 6.5. Then, as one of the main contributions of this chapter, our computational results in Section 6.6 compare the different presolving methods and their combination. The results show that, for the considered instances, presolving in the root node has limited effect, but node presolving – and bound tightening in particular – can result in a significant speed-up of up to 22 % in comparison to no presolving. Moreover, on one hand, presolving has a different impact on different types of instances. On the other hand, since the methods only take a negligible amount of time, they can easily be applied without much overhead. In conclusion, the techniques investigated in this chapter provide a very good basis for future applications of generic MISDP.

## 6.1 Presolving and MISDPs – An Overview

We start with a brief review of the three main techniques for solving (6.1):

1. *SDP-based branch-and-bound:* One can adapt the general nonlinear branch-and-bound process, as already proposed by Dakin [58] in 1965, by branching

on fractional variables and solving SDPs in each node. Two of the first solvers based on this idea are YALMIP [160] and SCIP-SDP [220], see the next paragraph for details on SCIP-SDP, and Gally et al. [112] for an analysis of subproblem properties in the tree.

2. *LP-based branch-and-bound:* The second technique was proposed by Sherali and Fraticelli [223], see also Krishnan and Mitchell [148]. It applies a linear programming based cutting-plane algorithm for solving the continuous subproblems in each node of the tree while branching on fractional variables, see the subsequent paragraph for more details. This LP-based approach is also implemented in SCIP-SDP (see Mars [173] and Gally [110] for computational results) and YALMIP. A corresponding convergence analysis was performed by Kobayashi and Takano [145].
3. *Outer approximation:* Outer approximation, proposed by Duran and Grossmann [80], was investigated for mixed-integer conic problems by Lubin et al. [164] and is implemented in the solver Pajarito [54]. We will not investigate this approach in this chapter, but will present results for the first two.

**Notes on SCIP-SDP** All the techniques that will be presented in this chapter, have been implemented in version 4.0 of the solver SCIP-SDP, which is a framework for solving mixed-integer semidefinite programs of the form (6.1). It is publically available at `https://wwwopt.mathematik.tu-darmstadt.de/scipsdp/` and is based on SCIP, available at `https://scipopt.org/`.

SCIP-SDP was initiated by Sonja Mars and Lars Schewe, see Mars [173], and then continued by Gally et al. [112] and Gally [110]. It features interfaces to the SDP solvers DSDP, MOSEK [181], and SDPA [254]. Major work has been put into the improvement of SCIP-SDP by Marc E. Pfetsch and the author of this thesis during the preparation of the material on presolving for MISDPs. In particular, all of the presolving routines that are introduced in this chapter have been implemented. On top of that, several further changes and additions, which are not connected to presolving have been conducted, which led to version 4.0 of SCIP-SDP. SCIP-SDP 4.0 contains about 50 000 lines of C-code, most of which have been touched since the last version 3.2. For a description of the many further changes and improvements for SCIP-SDP 4.0, see the corresponding section in the release report of SCIP 8 [25]. We remark that we slightly relaxed the feasibility and optimality tolerances in SCIP-SDP 4.0. Table 6.1 shows a comparison between SCIP-SDP 3.2 and 4.0 on the same testset as used by Gally et al. [112] which consists of 194 instances. Reported are the number of optimally solved instances, as well as the shifted geometric means of the number of processed nodes and the CPU time in seconds. We

**Table 6.1.** Performance comparison of SCIP-SDP 4.0 vs. SCIP-SDP 3.2 on a testset of 194 instances.

| | #opt | #nodes | time |
|---|---|---|---|
| SCIP-SDP 3.2 | 185 | 617.3 | 42.9 |
| SCIP-SDP 4.0 | 187 | 497.3 | 26.6 |

use MOSEK 9.2.40 [181] for solving the continuous SDP relaxations. The tests were performed on a Linux cluster with 3.5 GHz Intel Xeon E5-1620 Quad-Core CPUs, having 32 GB main memory and 10 MB cache. All computations were run single-threaded and with a time limit of one hour. The conclusion from these results is that SCIP-SDP 4.0 has significantly improved since the last version. In the remaining parts of this chapter, our implementation always refers to SCIP-SDP.

**Notes on the LP-based Approach** When solving general MISDPs of the form (6.1) with a cutting-plane approach, the positive semidefiniteness of $A(y)$ needs to be enforced through linear cuts. To do so, it is possible to use the following characterization of positive semidefiniteness:

$$\sum_{k=1}^{m} A^k y_k - A^0 \succeq 0 \quad \Leftrightarrow \quad v^\top A(y)\, v = v^\top \Big( \sum_{k=1}^{m} A^k y_k - A^0 \Big) v \geq 0 \quad \forall\, v \in \mathbb{R}^n.$$

Thus, if a given relaxation solution $y^*$ violates the SDP constraint $A(y^*) \succeq 0$, there exists $v^* \in \mathbb{R}^n$ with $(v^*)^\top A(y^*)\, v^* < 0$. Consequently, the valid linear inequality

$$(v^*)^\top \Big( \sum_{k=1}^{m} A^k y_k - A^0 \Big) v^* \geq 0$$

cuts the relaxation solution $y^*$ off. These cuts are sometimes called *eigenvector cuts* or *eigencuts*, since a simple choice for $v^*$ is an eigenvector for the smallest eigenvalue of $A(y^*)$, which is negative if the SDP constraint is violated. Of course, it is also possible to directly add several eigenvector cuts, for example, one for each negative eigenvalue of $A(y^*)$.

In SCIP-SDP, there are two possibilities to add eigenvector cuts. The first variant *separates* eigenvector cuts during the solution of the LP relaxation, that is, eigenvector cuts are added whenever a feasible solution of the LP relaxation does not satisfy the positive semidefiniteness constraint. This setting will be denoted by `LPA` in our experiments. The second variant, denoted by `LPE`, only *enforces* eigenvector cuts, that is, these cuts are only added, if an optimal solution of the LP relaxation satisfies the integrality constraints, but still violates the positive semidefiniteness constraint (a "lazy-cut" approach).

Although it is not the focus of this chapter, let us comment on the computations by Kobayashi and Takano [145] who compare the SDP-based approach in SCIP-SDP with their own implementation of two algorithms using an LP-based approach. The best performing method in [145] is to use LP relaxations in which eigenvalue cuts are only generated if all integer variables attain integral values (the lazy-cut approach). This method is quite similar to our method of only enforcing integral solutions (`LPE-MIX2`), see Section 6.6. The results of our computations differ in several aspects from [145]: For the LP-based approach, it is faster to also separate eigenvector cuts for fractional solutions and not only for integer valued solutions. Our implementation based on SDP relaxations is much faster on average than the LP-based approach. Note that in [145] an older version of SCIP-SDP with DSDP was used on the NEOS server. Here, we compare on the same machines, use an improved implementation, and use MOSEK as an SDP solver. Moreover, we test on similar but larger instances compared to [145], see Section 6.6.1.

Finally, let us remark that an approximation of SDPs by linear inequalities may need to be very large in the worst case, as shown in the next corollary by combining results from the literature. This is in contrast to second-order cone programs, for which $\varepsilon$-approximate extended formulations of polynomial size in the input and $\log(1/\varepsilon)$ exist, see Ben-Tal and Nemirovski [22].

**Corollary 6.2.** *There are SDPs of dimension $n \times n$ for which any polyhedral approximation is of size $2^{\Omega(n)}$.*

*Proof sketch.* Braun et al. [32] proved that one may need polyhedral extended formulations with extension complexity $2^{\Omega(n)}$ to construct tight approximations of the feasible regions of SDPs in $\mathbb{R}^{n\times n}$. The proof is based on constructing instances whose nonnegative rank is $2^{\Omega(n)}$. Moreover, as shown by Braun et al. [33], the nonnegative rank deviates from the minimal number of inequalities in a polyhedral description in the original dimension $n \times n$ by at most 1. □

## Literature Overview

We first note that SDP relaxations can be preprocessed to improve their numerical stability, for example by facial reduction techniques, see, e.g., Permenter and Parrilo [198, 199], Permenter et al. [200]. However, such features so far have neither been implemented into the SDP solver MOSEK, which we use in our computations, nor in our code.

In the following literature review, we concentrate on presolving techniques for problems containing integer variables, since this is the main focus of this chapter.

For MIPs, many presolving methods are known, see for instance Brearley et al. [34] and Crowder et al. [57]. We note that details are not needed for understanding the contributions of this chapter. We will, however, add some pointers to MIP-presolving techniques later and refer to the following literature for more information. An overview and new techniques were presented by Savelsbergh [217] and for a more recent overview, see Mahajan [168]. Achterberg [1] discusses the implementation of presolving in detail. Further recent contributions are introduced in Achterberg et al. [4] and Gemander et al. [117]. The last three publications describe the methods implemented in the framework SCIP. Presolving is even more important for MINLPs, see, e.g., Vigerske [249], Belotti et al. [20], Vigerske and Gleixner [250], and Puranik and Sahinidis [205].

Several presolving methods for MISDPs have been proposed by Mars [173], Gally et al. [112], and Gally [110]; we explain the most relevant ones in the following. Beyond the mentioned references, we are not aware of any other presolving techniques for MISDPs.

## Standard Presolving

Several known presolving steps are (relatively) straightforward to perform. For instance, possibly present linear inequalities in (6.1) can be presolved as for MIPs (see above for references). The following basic MISDP-specific methods have been introduced by Mars [173, Section 3.3.2] and partly extended by Gally [110]: Fixed variables can be removed by appropriately adjusting the constant matrix $A^0$. Similarly, (multi-)aggregated variables, i.e., variables that affinely depend on other variables, can be substituted, possibly adjusting the affected matrices $A^k$, $k \in [m]_0$. Furthermore, one can check whether all matrices $A^k$ for $k \in [m]_0$ are diagonal. In this case, the SDP constraint $A(y) \succeq 0$ can be replaced by corresponding linear inequalities. All these steps are automatically performed in our implementation.

Further presolving steps treat rather rare cases and are therefore not implemented: Zero matrices $A^k$ and their corresponding variables $y_k$ can be removed. Moreover, duplicate constraints $A(y) \succeq 0$ or duplicated blocks within $A(y) \succeq 0$ can be detected and removed. Redundant constraints $A(y) \succeq 0$ can be detected in several special cases, e.g., if all variables are binary, all $A^k$, $k \in [m]$, are positive semidefinite and $A^0$ is negative semidefinite. If $m = 1$ in the SDP constraint $A(y) \succeq 0$, i.e., there is only one variable, the feasible region is an interval (see Section 6.4.2); thus, the SDP constraint can be removed and the variable bounds can be adjusted. Furthermore, if all matrices $A^k$, for $k \in [m]_0$, contain the same 0 rows and columns, the dimension can be reduced. This last step is automatically performed in our implementation, each time an instance is passed to an SDP solver. Furthermore, Mars [173, Section 3.3.2] discusses methods to detect block structures in the SDP constraint. Under certain conditions, one can also apply dual presolving. For example, if for some $k \in [m]$ the matrix $A^k$ is positive semidefinite and disjoint from the rest (i.e., $A^k$ has no common nonzero with the other matrices), one can fix $y_k$ to its upper or lower bound, depending on the objective coefficient.

More expensive presolving includes so-called probing, see, e.g., Savelsbergh [217]. Probing tentatively fixes binary variables to 0 and 1 and then checks whether propagation of variable bounds leads to infeasibilities. If this happens, one can fix the binary variable to the opposite value. Moreover, implications between binary variables can be detected. Probing is automatically performed in our implementation, but the propagation methods often do not seem to be strong enough to allow for many probing reductions. One further method is *optimality based bound tightening* (OBBT) in which one maximizes/minimizes variables over a relaxation of the problem to determine lower and upper variable bounds, see, e.g., Gleixner et al. [120] for a recent variant. This method was adapted for MISDPs by Gally [110] and usually reduces the number of nodes in the tree, but increases running times. It is therefore not considered in the following. Dual fixing is a node presolving method, which is a generalization of reduced cost fixing, and is always used in our implementation, see Gally et al. [112].

We finally note that node presolving has secondary effects. For instance, it affects conflict analysis, which in this context summarizes techniques that derive so-called conflict constraints, i.e., linear, set covering or more general disjunctive constraints, based on the information that a certain node in the branch-and-bound tree is infeasible. We refer to Achterberg [2], Witzig [251] and Witzig et al. [252] for more information. If SDP relaxations are used, the generated conflicts only arise from the so-called conflict graph analysis, which applies if the infeasibility of the node has been determined by propagation of variable bounds. Preliminary computations showed that this does not significantly change the performance. In the LP-based approach, however, conflict analysis also uses LP infeasibility proofs and seems to have a negative impact, see the results in Section 6.6.

## 6.2 Linear Inequalities Implied by the SDP Relaxation

The following inequalities are known from the literature and can be added to (6.1) as linear inequalities. All these inequalities are implied by the SDP relaxation of (6.1), but might be useful for standard presolving with respect to linear constraints or when solving a linear relaxation.

- Mars [173, Section 3.3.2] observed that the constraint $A(y) \succeq 0$ implies that the diagonal entries of $A(y)$ are nonnegative (*Diagonal Greater equal Zero, DGZ*), i.e., for all $i \in [n]$:

$$\sum_{k=1}^{m} (A^k)_{ii}\, y_k - (A^0)_{ii} \geq 0. \tag{DGZ}$$

- If $A^k_{ii} = A^k_{jj} = 0$ for all $k \in [m]$ and $A^0_{ii}\, A^0_{jj} \geq 0$ for some $i \neq j \in [n]$, then the following inequality based on products of $2 \times 2$ minors (*2-Minor Product, 2MP*) is valid, see Gally [110, Prop. 5.11]:

$$\sum_{k=1}^{m} A^k_{ij}\, y_k \geq A^0_{ij} - \sqrt{A^0_{ii}\, A^0_{jj}}. \tag{2MP}$$

  Furthermore, if exactly one $A^k_{ij} \neq 0$, then this yields upper or lower bounds for the corresponding variable $y_k$, depending on the sign of $A^k_{ij}$. Further similar inequalities can be found in Gally [110, Prop. 5.13].

We also obtain the following slight generalization of the "diagonal-zero-implication cuts (DZI)" introduced by Gally [110], based on an observation of Mars [173]. These inequalities build on the presence of integral variables.

**Lemma 6.3.** *Let $i, j \in [n]$ with $i \neq j$ and $A^0_{ij} \neq 0$ as well as $A^0_{ii} \geq 0$. If $A^k_{ij} = 0$ for all $k \in [m]$, $A^k_{ii} = 0$ for all continuous variables $k \in [m] \setminus I$, and $\ell_k \geq 0$ for all integer variables $k \in I$, the following inequality is valid:*

$$\sum_{\substack{k \in I:\\ A^k_{ii} > 0}} y_k \geq 1. \tag{DZI}$$

*Proof.* Any $y$ feasible for (6.1) satisfies $A(y) \succeq 0$ and therefore also $A(y)_{ii} \geq 0$ as well as $A(y)_{jj} \geq 0$. The $2 \times 2$ minor w.r.t. $i$, $j$ yields $A(y)_{ii} \cdot A(y)_{jj} - (A(y)_{ij})^2 \geq 0$. By assumption $A(y)_{ij} = A^0_{ij} \neq 0$. This implies that $A(y)_{ii} \cdot A(y)_{jj} > 0$ and therefore $A(y)_{ii} > 0$ (and $A(y)_{jj} > 0$). Since $A^k_{ii} = 0$ for all $k \in [m] \setminus I$ and $\ell_k \geq 0$ for all $k \in I$, we obtain:

$$0 < A(y)_{ii} = \sum_{k=1}^{m} A^k_{ii}\, y_k - A^0_{ii} = \sum_{k \in I} A^k_{ii}\, y_k - A^0_{ii} \leq \sum_{\substack{k \in I:\\ A^k_{ii} > 0}} A^k_{ii}\, y_k - A^0_{ii}.$$

Since $A^0_{ii} \geq 0$, this implies that at least one variable $y_k$ with $k \in I$ and $A^k_{ii} > 0$ has to be positive, i.e., at least 1. □

Another family of valid inequalities are the so-called *2-Minor Linear Constraints (2ML)*, which are a special case of eigenvector cuts (see Section 6.1). For a positive semidefinite matrix $Z \succeq 0$, these inequalities are given by

$$Z_{ii} + Z_{jj} - 2\,Z_{ij} \geq 0, \tag{6.3}$$

$$Z_{ii} + Z_{jj} + 2\,Z_{ij} \geq 0. \tag{6.4}$$

They are obtained by restricting to the $2 \times 2$ minor w.r.t. $i$ and $j$ and multiplying from left and right by $(1, -1)^\top$ and $(1, 1)^\top$, respectively. If $Z = A(y)$, we obtain for the first inequality

$$\begin{aligned} &\sum_{k=1}^m A^k_{ii}\, y_k - A^0_{ii} + \sum_{k=1}^m A^k_{jj}\, y_k - A^0_{jj} - 2\Big(\sum_{k=1}^m A^k_{ij}\, y_k - A^0_{ij}\Big) \geq 0 \\ \Leftrightarrow \quad &\sum_{k=1}^m \Big(A^k_{ii} + A^k_{jj} - 2\,A^k_{ij}\Big)\, y_k \geq A^0_{ii} + A^0_{jj} - 2A^0_{ij}, \end{aligned} \tag{2ML}$$

and similarly for the second inequality. As above, these inequalities are implied by the SDP constraint $A(y) \succeq 0$, but might be used for propagation. Of course, any other eigenvector cut is also a valid inequality, but constraints (2ML) are relatively easy and sparse, at least in primal form (6.3) and (6.4).

## 6.3 Presolving Techniques Based on 2 × 2 minors

In this section, we develop methods that are based on taking $2 \times 2$ minors of a positive semidefinite matrix.

### Using Bounds on the Diagonal

**Lemma 6.4.** *Consider $Z \succeq 0$ with $0 \leq Z_{ii} \leq U_{ii}$ for all $i \in [n]$. Then*

$$-\sqrt{U_{ii}\, U_{jj}} \leq Z_{ij} \leq \sqrt{U_{ii}\, U_{jj}} \tag{6.5}$$

*holds for all $i$, $j \in [n]$.*

*Proof.* Since $Z$ is positive semidefinite, we have $Z_{ii}Z_{jj} - Z^2_{ij} \geq 0$. Rewriting this inequality yields $Z^2_{ij} \leq Z_{ii}Z_{jj} \leq U_{ii}\, U_{jj}$. Taking the square root gives the claim. □

**Remark 6.5.**

- The bounds in Lemma 6.4 are tight, even for a rank-1 matrix $Z$: consider the rank-1 all-ones matrix.
- Inequality (6.3) yields $Z_{ij} \leq \frac{1}{2}(Z_{ii} + Z_{jj}) \leq \frac{1}{2}(U_{ii} + U_{jj})$. This derived bound is dominated by (6.5), because

  $$Z_{ij} \leq \sqrt{U_{ii} \cdot U_{jj}} \leq \tfrac{1}{2}\big(U_{ii} + U_{jj}\big),$$

  using the inequality between the arithmetic and geometric mean.

Lemma 6.4 can partly be translated to the matrix pencil format $A(y)$ by defining

$$\tilde{U}_{ij} \coloneqq \sum_{k \in [m] : A^k_{ij} > 0} A^k_{ij}\, u_k + \sum_{k \in [m] : A^k_{ij} < 0} A^k_{ij}\, \ell_k - A^0_{ij}. \tag{6.6}$$

Thus, for any $\ell \leq y \leq u$, we have $A(y)_{ij} \leq \tilde{U}_{ij}$. This directly yields the following lemma.

**Lemma 6.6.** *For any solution $y \in \mathbb{R}^m$ of* (6.1)*, we have*

$$-\sqrt{\tilde{U}_{ii}\, \tilde{U}_{jj}} \leq A(y)_{ij} \leq \sqrt{\tilde{U}_{ii}\, \tilde{U}_{jj}} \tag{6.7}$$

*for all $i$, $j \in [n]$.*

The downside of Inequalities (6.7) is that they can be quite weak if $A(y)_{ij}$ depends on many variables. We therefore concentrate on the case in which each entry $A(y)_{ij}$ depends on one variable only, that is, there exists $k = k(i,j) \in [m]$ such that $A(y)_{ij} = A^k_{ij}\, y_k - A^0_{ij}$ with $A^k_{ij} \neq 0$. In this case, Inequalities (6.7) are equivalent to

$$\frac{-\sqrt{\tilde{U}_{ii}\, \tilde{U}_{jj}} + A^0_{ij}}{A^k_{ij}} \leq y_k \leq \frac{\sqrt{\tilde{U}_{ii}\, \tilde{U}_{jj}} + A^0_{ij}}{A^k_{ij}}, \tag{PropUB}$$

if $A^k_{ij} > 0$ and similarly if $A^k_{ij} < 0$. If $k \in I$, i.e., variable $y_k$ is integral, the lower bound can be rounded up and the upper bound down. In our implementation, these inequalities are used in presolving and possibly for propagation of variable bounds in every node, which is denoted by *Propagate Upper Bounds (PropUB)*. Again, since Inequalities (PropUB) are valid for the SDP relaxation, integral variables have to be present or a linear relaxation has to be solved in order for Inequalities (PropUB) to be computationally useful.

By using trace constraints, one can also compute different bounds on the off-diagonal elements as follows; this slightly strengthens [111, Lemma 1].

**Lemma 6.7.** *Consider $Z \succeq 0$ with $\operatorname{tr}(Z) \leq \alpha$. Then*

$$-\tfrac{\alpha}{2} \leq Z_{ij} \leq \tfrac{\alpha}{2} \tag{6.8}$$

*holds for all $i$, $j \in [n]$ with $i \neq j$.*

*Proof.* Since $Z \succeq 0$, again we have $Z_{ij}^2 \leq Z_{ii} Z_{jj}$. By the trace constraint and the fact that the diagonal entries are nonnegative, $Z_{ii} + Z_{jj} \leq \alpha$. Moreover, we obtain:

$$Z_{ii}\, Z_{jj} \leq Z_{ii}(\alpha - Z_{ii}) = \alpha Z_{ii} - Z_{ii}^2.$$

Taking the derivative and equating 0 yields a maximal point $Z_{ii}^\star = \frac{\alpha}{2}$. Consequently,

$$Z_{ij}^2 \leq Z_{ii}\, Z_{jj} \leq \alpha Z_{ii}^\star - (Z_{ii}^\star)^2 = \tfrac{\alpha^2}{2} - \tfrac{\alpha^2}{4} = \tfrac{\alpha^2}{4}.$$

Taking the square root shows the claim. □

Inequalities (6.8) can again be transferred to $A(y) \succeq 0$, but with the same disadvantages. Therefore, we only use these inequalities in the case that $A(y)_{ij}$ only depends on a single variable. As before, integrality of variables can be exploited for rounding the bounds.

## Using Bounds on the Off-Diagonal

We now derive affine inequalities that depend on $2 \times 2$ minors. The following result is motivated by and generalizes the special case in Nohra et al. [189].

**Lemma 6.8.** *Consider a positive semidefinite matrix $Z \in \mathcal{S}_+^n$ with $L \leq Z \leq U$, where the inequalities are meant componentwise. Then for all $i$ and $j \in [n]$:*

$$U_{jj}\, Z_{ii} \geq 2\, L_{ij}\, Z_{ij} - L_{ij}^2 \qquad \text{and} \qquad U_{jj}\, Z_{ii} \geq 2\, U_{ij}\, Z_{ij} - U_{ij}^2. \tag{6.9}$$

*Proof.* We first obtain

$$(Z_{ij} - L_{ij})^2 \geq 0 \quad \Leftrightarrow \quad Z_{ij}^2 \geq 2\, L_{ij}\, Z_{ij} - L_{ij}^2.$$

The $2 \times 2$ minor for $i$ and $j$ gives $Z_{jj}\, Z_{ii} - Z_{ij}^2 \geq 0$. Together with $Z_{ii} \geq 0$, this yields

$$2\, L_{ij}\, Z_{ij} - L_{ij}^2 \leq Z_{ij}^2 \leq Z_{jj}\, Z_{ii} \leq U_{jj}\, Z_{ii}.$$

The second inequality arises similarly. □

**Remark 6.9.**

- Inequalities (6.9) are implied by the SDP constraint, so that they can only be useful when solving LPs or for integral variables. Moreover, assume that $L_{ij} < 0$ and $U_{ij} > 0$, which is typical for $i \neq j$. Then these inequalities are nontrivial, that is, the right-hand-side is nonnegative, if $Z_{ij} \leq L_{ij}/2$ and $Z_{ij} \geq U_{ij}/2$, respectively.
- Note that cuts like (6.3) or (6.4) do not take the lower and upper bounds into account. Thus, Inequalities (6.9) might further strengthen an LP relaxation.
- However, if we use $Z_{ii} \leq U_{ii}$, the last inequality in (6.9) yields (if $U_{ij} > 0$):

  $$Z_{ij} \leq \frac{U_{jj}\, U_{ii} + U_{ij}^2}{2\, U_{ij}}. \tag{6.10}$$

  The right hand-side is stronger than $Z_{ij} \leq U_{ij}$ if $U_{ii}\, U_{jj} \leq U_{ij}^2$. If $U$ is positive semidefinite, this never happens. Thus, Inequalities (6.9) are preferable over (6.10).

We transfer Inequalities (6.9) to the form $A(y) \succeq 0$ as in Lemma 6.6. For the second inequality in (6.9), we obtain:

$$\begin{aligned} & 2\, \tilde{U}_{ij}\, A(y)_{ij} - \tilde{U}_{jj}\, A(y)_{ii} \leq \tilde{U}_{ij}^2 \\ \Leftrightarrow \quad & \sum_{k=1}^{m} 2\, \tilde{U}_{ij}\, A_{ij}^k\, y_k - \sum_{k=1}^{m} \tilde{U}_{jj}\, A_{ii}^k\, y_k \leq \tilde{U}_{ij}^2 + \sum_{k=1}^{m} 2\, \tilde{U}_{ij}\, A_{ij}^0 - \sum_{k=1}^{m} \tilde{U}_{jj}\, A_{ii}^0, \end{aligned} \tag{2MV}$$

where $\tilde{U}_{ij}$ and $\tilde{U}_{jj}$ are defined as in (6.6). These inequalities are referred to as *2-Minor Variable Bounds (2MV)*.

A particular case in which Inequalities (6.9) might be useful arises in SDP relaxations of quadratic programs, see Lovász and Schrijver [162] or Luo et al. [165], or in truss topology optimization, see (TTD). A quadratic program in the variable $x$ can be transformed into an SDP by introducing a matrix variable $X$ and replacing $X = xx^\top$. Thus, a quadratic term $xQx^\top$ with a matrix $Q$ is equivalent to $\langle Q, X \rangle_{\mathrm{F}}$, which is linear in $X$. In order to obtain an SDP, the exact nonconvex equality $X = xx^\top$ is relaxed to $X - xx^\top \succeq 0$. By using the Schur complement, this is equivalent to

$$\begin{pmatrix} t & x^\top \\ x & X \end{pmatrix} \succeq 0.$$

**Corollary 6.10.** *Consider* $(X, x, t) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n \times \mathbb{R}$ *satisfying*

$$\begin{pmatrix} t & x^\top \\ x & X \end{pmatrix} \succeq 0, \quad \ell \le x \le u, \quad \ell^{(t)} \le t \le u^{(t)},$$

*where $t$ is a scalar variable. Then for all $i \in [n]$:*

$$u^{(t)}\, X_{ii} \ge 2\,\ell_i\, x_i - \ell_i^2 \qquad \textit{and} \qquad u^{(t)}\, X_{ii} \ge 2\, u_i\, x_i - u_i^2.$$

## 6.4 Bound Tightening Based on SDP Constraints

In this section, we investigate how SDP constraints $A(y) \succeq 0$ can be used to tighten variable bounds. For an index $k \in [m]$, define

$$P_k \coloneqq \{i \in [m] \setminus \{k\} \,:\, A^i \succeq 0\}, \qquad N_k \coloneqq \{i \in [m] \setminus \{k\} \,:\, A^i \preceq 0\},$$

as well as

$$\underline{\mu}_k \coloneqq \begin{cases} \inf \Big\{\mu \,:\, A^k\,\mu + \sum\limits_{i\in P_k} A^i\, u_i + \sum\limits_{j\in N_k} A^j\,\ell_j - A^0 \succeq 0\Big\} & \text{if } \begin{array}{ll} u_i < \infty, & i \in P_k, \\ \ell_i > -\infty, & i \in N_k, \end{array} \\ -\infty & \text{otherwise,} \end{cases} \tag{6.11}$$

$$\overline{\mu}_k \coloneqq \begin{cases} \sup \Big\{\mu \,:\, A^k\,\mu + \sum\limits_{i\in P_k} A^i\, u_i + \sum\limits_{j\in N_k} A^j\,\ell_j - A^0 \succeq 0\Big\} & \text{if } \begin{array}{ll} u_i < \infty, & i \in P_k, \\ \ell_i > -\infty, & i \in N_k, \end{array} \\ +\infty & \text{otherwise.} \end{cases} \tag{6.12}$$

Both $\underline{\mu}_k$ and $\overline{\mu}_k$ might be ±∞, even if all bounds are finite, for instance, if $A^k$ is negative or positive definite, respectively. Moreover, both might be simultaneously finite. The two SDPs in (6.11) and (6.12) only contain a single variable and can be solved with the technique discussed in Section 6.4.2 below.

The following lemma shows that the lower or upper bounds of the variables can be tightened, depending on the semidefiniteness of the coefficient matrices. This procedure is denoted by `TB` in our experiments.

**Lemma 6.11** (Tighten Bounds (`TB`))**.** *Let all $A^k$, $k \in [m]$, be positive or negative semidefinite. Then, $A(y) \succeq 0$ implies that $\underline{\mu}_k \le y_k \le \overline{\mu}_k$ for all $k \in [m]$. Finite bounds can be rounded for integral variables.*

*Proof.* Suppose that $y_k < \underline{\mu}_k$ or $y_k > \overline{\mu}_k$. Then, by definition of $\underline{\mu}_k$ and $\overline{\mu}_k$, there exists $x \in \mathbb{R}^n$ with

$$\begin{aligned}
0 &> x^\top \Big(A^k\, y_k + \sum_{i \in P_k} A^i\, u_i + \sum_{i \in N_k} A^i\, \ell_i - A^0\Big) x \\
&= x^\top A^k x\, y_k + \sum_{i \in P_k} \underbrace{x^\top A^i x}_{\geq 0}\, u_i + \sum_{i \in N_k} \underbrace{x^\top A^i x}_{\leq 0}\, \ell_i - x^\top A^0 x \\
&\geq x^\top A^k x\, y_k + \sum_{i \in P_k} x^\top A^i x\, y_i + \sum_{i \in N_k} x^\top A^i x\, y_i - x^\top A^0 x \\
&= x^\top \Big(\sum_{i=1}^{m} A^i\, y_i - A^0\Big) x,
\end{aligned}$$

which is a contradiction to $A(y) \succeq 0$. Thus, $\underline{\mu}_k \leq y_k \leq \overline{\mu}_k$. □

**Remark 6.12.**

- The conditions of Lemma 6.11 are frequently fulfilled for instances that we consider later in the numerical experiments in Section 6.6; namely for 75 out of 185 instances in our testset, all matrices $A^k$ are positive semidefinite, see Section 6.6.1. If some matrices are indefinite, a derivation of valid bounds is currently unknown.
- It is easy to include lower and upper bounds $\ell_k$ and $u_k$, respectively, into bound tightening by adding the constraint $\ell_k \leq \mu \leq u_k$ to (6.11) and (6.12). This makes the problems bounded if the bounds are finite, see Section 6.4.2.
- If all $A^k$, $k \in [m]_0$, are diagonal matrices, $A(y) \succeq 0$ specializes to a linear inequality $a^\top y - a_0 \geq 0$ with $a \in \mathbb{R}^m$ and $a_0 \in \mathbb{R}$. If $a_k > 0$, we obtain

$$y_k \geq \mu_k = \frac{1}{a_k}\Big(a_0 - \sum_{\substack{i : a_i > 0 \\ i \neq k}} a_i\, u_i - \sum_{j : a_j < 0} a_j\, \ell_j\Big),$$

  which is linear bound tightening, i.e., Lemma 6.11 generalizes the linear case.
- We note that Inequalities (6.5) are implied by Lemma 6.11. This can be seen as follows: Assume that we have a matrix $Z \succeq 0$ with some finite lower bounds $L \in \mathbb{R}^{n \times n}$ (the exact values are not important, but they make (6.12) finite). Write $Z = \sum_{i,j=1}^{n} E^{ij}\, Z_{ij} \succeq 0$, where $E^{ij} \in \mathbb{R}^{n \times n}$ is 0 except for positions $(i, j)$ and $(j, i)$, where it is 1. Then the optimal value $\bar{\mu}$ of (6.12) for variable $Z_{ij}$ yields that the $2 \times 2$ minor for $i$ and $j$ is nonnegative, i.e., $U_{ii}\, U_{jj} - \bar{\mu}^2 \geq 0$, which is (6.5). In comparison to the bounds of Lemma 6.11, the ones in (6.5) (or (6.7)) can be computed more efficiently and depend on fewer variable bounds.

### 6.4.1 Convergence of Bound Tightening

Lemma 6.11 can be applied iteratively and we investigate the convergence of this process. We assume that all coefficient matrices $A^k$, $k \in [m]$, are positive or negative semidefinite and that the initial bounds $\ell^{(0)}$ and $u^{(0)}$ are finite for all variables. Moreover, in each iteration, we incorporate the bounds $\ell$ and $u$ obtained in the previous iteration (or the initial bounds for the first iteration) in (6.11) and (6.12) by using $\max\{\ell, \underline{\mu}(\ell, u)\}$ and $\min\{u, \overline{\mu}(\ell, u)\}$. Thus, (6.11) and (6.12) always yield finite bounds. The analysis in this section uses similar arguments as Belotti et al. [19].

Let $\underline{\mu}(\ell, u)$ and $\overline{\mu}(\ell, u) \in \mathbb{R}^m$ be the lower and upper bounds derived from Lemma 6.11 for each variable, where the constraint $\ell_k \leq \mu \leq u_k$ is incorporated into (6.11) and (6.12). Define the interval set $\mathcal{I} := \{(\ell, u) \in \mathbb{R}^n \times \mathbb{R}^n \,:\, \ell \leq u\}$ with the following ordering for $(\ell, u)$, $(\ell', u') \in \mathcal{I}$:

$$(\ell, u) \leq_{\mathcal{I}} (\ell', u') \quad \Leftrightarrow \quad \ell' \leq \ell,\ u \leq u'.$$

Thus, if $(\ell, u) \leq_{\mathcal{I}} (\ell', u')$, then the bounds $(\ell, u)$ are at least as tight as $(\ell', u')$. Let

$$F \colon \mathcal{I} \to \mathcal{I},\ (\ell, u) \mapsto \big(\max\{\ell, \underline{\mu}(\ell, u)\}, \min\{u, \overline{\mu}(\ell, u)\}\big),$$

where min/max is applied componentwise. Thus, $F$ represents one step of bound tightening according to Lemma 6.11, making sure that the bounds do not get weaker.

**Lemma 6.13.** *$F$ is a contraction, i.e., $F(\ell, u) \leq_{\mathcal{I}} (\ell, u)$ for all $(\ell, u) \in \mathcal{I}$, and monotone, i.e., $(\ell, u) \leq_{\mathcal{I}} (\ell', u')$ implies $F(\ell, u) \leq_{\mathcal{I}} F(\ell', u')$.*

*Proof.* By definition of the max and min operations, $F$ is a contraction. For monotonicity, we concentrate on the upper bounds (the lower bounds are similar). Let $f(\ell, u) := \min\{u, \overline{\mu}(\ell, u)\}$ and similarly for $f(\ell', u')$. Assume for a contradiction that $(\ell, u) \leq_{\mathcal{I}} (\ell', u')$ (and thus $\ell' \leq \ell$, $u \leq u'$), but $\mu := f(\ell, u)_k > f(\ell', u')_k =: \mu'$ for some $k \in [m]$. Thus, the matrix $A^k\,\mu + \sum_{i \in P_k} A^i\, u'_i + \sum_{j \in N_k} A^j\, \ell'_j - A^0$ is not positive semidefinite by definition of $\mu'$. Therefore, there exists $x \in \mathbb{R}^n$ with

$$\begin{aligned}
0 &> x^\top \Big(A^k\,\mu + \sum_{i \in P_k} A^i\, u'_i + \sum_{i \in N_k} A^i\, \ell'_i - A^0\Big)x \\
&= x^\top A^k x\,\mu + \sum_{i \in P_k} \underbrace{x^\top A^i x}_{\geq 0}\, u'_i + \sum_{i \in N_k} \underbrace{x^\top A^i x}_{\leq 0}\, \ell'_i - x^\top A^0 x \\
&\geq x^\top A^k x\,\mu + \sum_{i \in P_k} x^\top A^i x\, u_i + \sum_{i \in N_k} x^\top A^i x\, \ell_i - x^\top A^0 x \\
&= x^\top \Big(A^k\,\mu + \sum_{i \in P_k} A^i\, u_i + \sum_{i \in N_k} A^i\, \ell_i - A^0\Big)x,
\end{aligned}$$

which is a contradiction to the last matrix in parentheses being positive semidefinite by definition of $\mu = \min\{u_k,\, \overline{\mu}(\ell,u)_k\}$. □

**Theorem 6.14.** *The operator $F$ has a unique greatest fixed point* $\operatorname{gfix}(F)$*, defined as* $\operatorname{gfix}(F) \coloneqq \sup\{(\ell,u) \in \mathcal{I} \,:\, (\ell,u) \leq_{\mathcal{I}} F(\ell,u)\}$.

*Proof.* Note that $\mathcal{I}$ forms a complete lattice. We always have $F(\ell,u) \leq_{\mathcal{I}} (\ell,u)$, since $F$ is a contraction. Thus, the interval set $\{(\ell,u) \in \mathcal{I} \,:\, (\ell,u) \leq_{\mathcal{I}} F(\ell,u)\}$ contains all fixed points. The result then follows by the Knaster-Tarski Theorem [234], see, e.g., Fritz [106, Theorem 20.4]. □

As in [19], we define the size $|(\ell,u)|$ of the interval $(\ell,u) \in \mathcal{I}$ as $\sum_{i=1}^m u_i - \ell_i$. Then [19] shows that $|\operatorname{gfix}(F)| \geq |(\ell,u)|$ for all fixed points $(\ell,u)$ of $F$. Thus, $\operatorname{gfix}(F)$ is the solution of

$$\max\left\{|(\ell,u)| \,:\, (\ell,u) \leq_{\mathcal{I}} F(\ell,u),\; (\ell,u) \leq_{\mathcal{I}} (\ell^{(0)},u^{(0)})\right\},$$

where $(\ell^{(0)},u^{(0)})$ denote the initial bounds. This can be written as the following SDP:

$$\begin{aligned} \max\quad & \sum_{i=1}^{m} u_i - \ell_i \\ \text{s.t.}\quad & A^k\,\ell_k + \sum_{i\in P_k} A^i\,u_i + \sum_{j\in N_k} A^j\,\ell_j - A^0 \succeq 0 \quad \forall k \in [m], \\ & A^k\,u_k + \sum_{i\in P_k} A^i\,u_i + \sum_{j\in N_k} A^j\,\ell_j - A^0 \succeq 0 \quad \forall k \in [m], \\ & \ell^{(0)} \leq \ell,\; u \leq u^{(0)},\; \ell \leq u. \end{aligned} \tag{6.13}$$

Let $(\ell^\star,u^\star)$ be an optimal solution of (6.13). Then this solution is a fixed point: By the constraints, we have $\ell^\star \geq \underline{\mu}$ and $u^\star \leq \overline{\mu}$. Thus, these bounds would not be tightened by $F$. Moreover, let $\{(\ell_k,u_k)\}$ be the sequence of bounds produced by iteratively applying $F$ as long as this changes some bounds. Since $F$ is monotone, the interval size $|(\ell_k,u_k)|$ is decreasing. Thus, the sequence $\{(\ell_k,u_k)\}$ will converge to the optimal solution $(\ell^\star,u^\star)$.

In our implementation, we iteratively apply Lemma 6.11 as long as this changes bounds of variables, instead of solving the SDP (6.13), because (6.13) is quite expensive to solve. Moreover, we can round bounds of integer variables after each iteration. Note that rounding for integer variables complicates the analysis of fixed points. Indeed, Bordeaux et al. [29] show that deciding the existence of an integral fixed point is NP-complete.

As we shall see, bound tightening is often successful deeper in the tree using bounds tightened by other components of the solver.

## 6.4.2 Computing Tightening Scalings

While in the linear case the values $\underline{\mu}_k$ and $\overline{\mu}_k$ can be computed easily, in the general case, it amounts to solving an SDP with one variable. For this, let us rewrite (6.11) and (6.12) in the presence of scalar lower and upper bounds $\ell$ and $u$, respectively, objective direction $\gamma \in \{\pm 1\}$, and appropriate $A$, $B \in \mathbb{R}^{n \times n}$ as

$$\mu^\star := \inf \{\gamma\, \mu \,:\, \mu\, A - B \succeq 0,\ \ell \leq \mu \leq u\}. \tag{6.14}$$

Problem (6.14) can be solved in different ways. In fact, there are several special cases in which (6.14) – with infinite bounds – is easy to solve, for instance, if $A = 0$ or $B = 0$. If $A$ is positive definite, there exists an invertible matrix $V$ with $V^\top A V = \mathbb{I}_n$, where $\mathbb{I}_n$ is the $n \times n$ identity matrix. It is then easy to see that $\mu^\star = \lambda_{\max}(V^\top B V)$, the maximal eigenvalue of $V^\top B V$. If there exists $\hat{\mu}$ with $\hat{\mu} A - B \succ 0$, Pong and Wolkowicz [203] as well as Jiang et al. [134] ([203] cites Lancaster and Rodman [152]) describe an algorithm based on Cholesky decomposition; these articles arise in the context of generalized trust region problems. In one final special case, the matrices $A$ and $B$ are simultaneously diagonizable. Then, there exists an invertible matrix $V$ with $V^\top(\mu A - B)V = \mu\, C - D$, where $C$ and $D$ are diagonal matrices. After computing this decomposition, Problem (6.14) is easy to solve.

Here, we are interested in the general case of Problem (6.14). Inspired by Strabić [231] and Higham et al. [129], we consider a semismooth Newton method. We state and prove the following for completeness.

**Lemma 6.15.** *Let $A$, $B \in \mathcal{S}^n$. Then, the function $f\colon \mathbb{R} \to \mathbb{R}$, $\mu \mapsto \lambda_{\min}(\mu\, A - B)$ is concave and hence continuous.*

*Proof.* For a symmetric matrix $C \in \mathcal{S}^n$, the variational characterization of the minimal eigenvalue reads $\lambda_{\min}(C) = \min\{x^\top C x \,:\, \|x\|_2 = 1\}$, see also (5.27). Thus, for $C$, $D \in \mathcal{S}^n$,

$$\begin{aligned} \lambda_{\min}(C + D) &= \min_{\|x\|_2 = 1}\ x^\top (C + D) x \\ &\geq \min_{\|x\|_2 = 1}\ x^\top C x + \min_{\|x\|_2 = 1}\ x^\top D x = \lambda_{\min}(C) + \lambda_{\min}(D). \end{aligned} \tag{6.15}$$

In order to show concavity of $f$, let $\alpha \in [0, 1]$ and $\mu_1$, $\mu_2 \geq 0$. Then, by using (6.15),

$$\begin{aligned} f\big(\alpha\mu_1 + (1-\alpha)\mu_2\big) &= \lambda_{\min}\big(\alpha(\mu_1\, A - B) + (1-\alpha)(\mu_2\, A - B)\big) \\ &\geq \lambda_{\min}\big(\alpha(\mu_1\, A - B)\big) + \lambda_{\min}\big((1-\alpha)(\mu_2\, A - B)\big) \\ &= \alpha\, f(\mu_1) + (1-\alpha)\, f(\mu_2). \end{aligned}$$

□

Lemma 6.15 implies that Problem (6.14) is convex. Moreover, if the optimal value of (6.14) is finite, it is attained: Otherwise, assume $\gamma = 1$ and that there exists a sequence $(\mu_k)$ of feasible points with $\mu_k \to \mu^\star$, where $\mu^\star$ is the value of (6.14). Since $f$ is continuous, we obtain $f(\mu_k) \to f(\mu^\star)$ and hence $f(\mu^\star) \geq 0$, i.e., $\mu^\star$ is feasible. Moreover, the following lemma shows how to obtain supergradients for (6.14).

**Lemma 6.16.** *Let $\hat{\mu} \in \mathbb{R}$ and $\hat{v}$ be a unit eigenvector for $\lambda_{\min}(\hat{\mu}\, A - B)$. Then $\hat{v}^\top A\hat{v}$ is a supergradient, i.e.,*

$$\lambda_{\min}(\mu\, A - B) \leq \lambda_{\min}(\hat{\mu}\, A - B) + (\mu - \hat{\mu})\, \hat{v}^\top A\hat{v}$$

*for all $\mu \in \mathbb{R}$. In particular, if $\hat{v}^\top A\hat{v} = 0$, then $\lambda_{\min}(\hat{\mu}\, A - B)$ is maximal.*

*Proof.* Since $\hat{v}$ is a unit eigenvector for $\lambda_{\min}(\hat{\mu}\, A - B)$, we have

$$\lambda_{\min}(\hat{\mu}\, A - B) = \hat{v}^\top(\hat{\mu}\, A - B)\hat{v}.$$

This implies

$$\begin{aligned} \hat{v}^\top(\mu\, A - B)\hat{v} &= \hat{v}^\top(\hat{\mu}\, A - B)\hat{v} + (\mu - \hat{\mu})\, \hat{v}^\top A\hat{v} \\ &= \lambda_{\min}(\hat{\mu}\, A - B) + (\mu - \hat{\mu})\, \hat{v}^\top A\hat{v}, \end{aligned}$$

where $\mu \in \mathbb{R}$. Using the variational characterization of the minimal eigenvalue yields $\lambda_{\min}(\mu\, A - B) \leq \hat{v}^\top(\mu\, A - B)\hat{v}$, since $\hat{v}$ has unit norm. This concludes the proof. □

Algorithm 3 provides the details of the resulting semismooth Newton method for solving Problem (6.14) for the case $\gamma = 1$; the algorithm for $\gamma = -1$ is very similar. Furthermore, the following considerations explain the crucial steps of Algorithm 3.

- In the case of Step 3, we use Lemma 6.16 for $\hat{\mu} = u$, $\hat{v} = w$ to get

  $$\lambda_{\min}(\mu\, A - B) \leq \underbrace{\lambda}_{<0} + \underbrace{(\mu - u)}_{\leq 0}\, \underbrace{\hat{v}^\top Av}_{>0} < 0$$

  for every $\mu$. Therefore, the problem is infeasible.
- In Step 7, if $\lambda > 0$, then $\mu = \ell$ is feasible and clearly the optimal solution.
- In Step 10, we have $\lambda < 0$ and $v^\top Av \leq 0$. Again using Lemma 6.16 for $\hat{\mu} = \ell$ and $\hat{v} = v$, we obtain

  $$\lambda_{\min}(\mu\, A - B) \leq \underbrace{\lambda}_{<0} + \underbrace{(\mu - \ell)}_{\geq 0}\, \underbrace{\hat{v}^\top Av}_{<0} < 0$$

  for all $\mu$, and the problem is infeasible.

- Step 14 computes $\mu_{k+1}$ such that $\lambda_k + (\mu_{k+1} - \mu_k)(v^k)^\top A v^k = 0$, i.e., the eigenvalue estimation via Lemma 6.16 becomes 0 (this is akin to the Newton iteration).
- Note that because of the while conditions, the sequence $(\mu_k)$ is strictly monotonously increasing.

**Remark 6.17.** We can apply general convergence theory, for instance, Theorem 7.5.3 in [93] (see also Qi and Sun [206]), which proves that the semismooth Newton method converges Q-superlinearly to a zero $\mu^\star$ of $f(\mu) = \lambda_{\min}(\mu\, A - B)$, given that the derivative $\partial f(\mu^\star) = (v^k)^\top A v^k$ is nonsingular and the starting point lies near $\mu^\star$. Since $f$ is concave, $f$ is semismooth and the theorem can be applied.

**Algorithm 3:** Semismooth Newton method

```
   Input: Matrices $A$ and $B$, scalar lower and upper bounds $\ell < u$
   Output: Solution of $\min\{\mu : \mu\, A - B \succeq 0,\ \ell \le \mu \le u\}$ or “infeasible”
 1 compute unit eigenvector $w$ for minimal eigenvalue $\lambda$ of $A\, u - B$;
 2 if $\lambda < 0$ and $w^\top A w > 0$ then
 3     return “infeasible”;
 4 end
 5 compute unit eigenvector $v$ for minimal eigenvalue $\lambda$ of $A\, \ell - B$;
 6 if $\lambda \ge 0$ then
 7     return $\ell$;
 8 end
 9 if $v^\top A v \le 0$ then
10     return “infeasible”;
11 end
12 $\mu_0 \leftarrow \ell$, $\lambda_0 \leftarrow \lambda$, $v_0 \leftarrow v$, $k \leftarrow 0$;
13 while $\lambda_k < 0$ and $(v^k)^\top A v^k > 0$ do
14     $\mu_{k+1} = \mu_k - \frac{\lambda_k}{(v^k)^\top A v^k}$;
15     if $\mu_{k+1} > u$ then
16         break;
17     end
18     compute unit eigenvector $v^{k+1}$ for minimal eigenvalue $\lambda_{k+1}$ of
        $A\, \mu_{k+1} - B$;
19     $k \leftarrow k + 1$;
20 end
21 if $\lambda_k < 0$ then
22     return “infeasible”
23 end
24 return $\mu_k$
```

As noted above, since we start with $\mu_0 = \ell$, after Steps 7 and 10, the sequence $(\mu_k)$ is strictly monotonously increasing. Therefore, the process always globally converges. However, if $\partial f(\mu_k)$ or $\partial f(\mu^\star)$ becomes singular, we cannot rely on Q-superlinear convergence.

## 6.5 Coefficient Tightening Based on SDP Constraints

Apart from tightening variable bounds, an SDP constraint $A(y) \succeq 0$ can also be used to scale individual matrices $A^k$, which is demonstrated in this section. In the linear case, tightening the coefficients of a linear inequality involving integer variables is used to reduce the number of fractional solutions to this constraint, whereas the feasible integral solutions remain unchanged. Moreover, it may lead to a stronger continuous relaxation. In order to tighten coefficients in an SDP constraint $A(y) \succeq 0$, define

$$\tilde{\mu}_k = \min\{\mu \,:\, A^k\,\mu - A^0 \succeq 0,\ \ell_k \leq \mu \leq u_k\}$$

and $\hat{\mu}_k = \min\{\tilde{\mu}_k, 1\}$ for $k \in [m]$. The following lemma describes a way to "tighten" matrices $A_k$, which we denote by `TM` in our experiments.

**Lemma 6.18** (Tighten Matrices (`TM`))**.** *Let $A^k \succeq 0$ for all $k \in [m]$, and let $\ell_k \geq 0$ for all $k \in [m]$. Furthermore, let $y_k \in \{0,1\}$ for all $k \in I$, i.e., assume all integer variables are binary variables. Then, for all $y \in \mathbb{R}^m$ with $\ell \leq y \leq u$:*

$$A(y) \succeq 0 \quad \Leftrightarrow \quad \sum_{k=1}^{m} \hat{\mu}_k\, A^k\, y_k - A^0 \succeq 0,$$

*where we define $\hat{\mu}_k = 1$ for $k \notin I$.*

*Proof.* First assume that $\sum_{k=1}^{m} \hat{\mu}_k\, A^k\, y_k - A^0 \succeq 0$. Since $A^k \succeq 0$ and $\ell_k \geq 0$ for all $k \in [m]$ by assumption, we get $0 \leq \hat{\mu}_k \leq 1$. Then for all $x \in \mathbb{R}^n$

$$0 \leq x^\top \Big( \sum_{k=1}^{m} \hat{\mu}_k\, A^k\, y_k - A^0 \Big) x = \sum_{k=1}^{m} \hat{\mu}_k\, \underbrace{x^\top A^k x\, y_k}_{\geq 0} - x^\top A^0 x \leq x^\top \Big( \sum_{k=1}^{m} A^k\, y_k - A^0 \Big) x,$$

which implies that $A(y) \succeq 0$.

We now assume that $A(y) \succeq 0$. By removing terms with $y_k = 0$ for $k \in I$, we can assume that $y_k = 1$ for all $k \in I$. Thus, $\sum_{k \in I} A^k + \sum_{k \notin I} A^k\, y_k - A^0 \succeq 0$. Since $A^k \succeq 0$ and $\ell_k \geq 0$ for all $k \in [m]$ by assumption,we get $0 \leq \hat{\mu}_k \leq 1$. If

$\hat{\mu}_k = 1$ for all $k \in I$ then the statement is directly clear, since $\hat{\mu}_k = 1$ for $k \notin I$ by definition. Therefore assume that there exists $k \in I$ with $\hat{\mu}_k = \tilde{\mu}_k < 1$. Without loss of generality assume further that $\hat{\mu}_j = 1$ for all $j \in I \setminus \{k\}$. But then already $\hat{\mu}_k \, A^k - A^0 \succeq 0$. Since $y_j \geq \ell_j \geq 0$ and $\hat{\mu}_j = 1$ for all $j \in [m] \setminus \{k\}$ by assumption, we have $\hat{\mu}_j A^j y_j \succeq 0$ for all $j \in [m] \setminus \{k\}$. Thus, adding these terms yields

$$\hat{\mu}_k \, A^k - A^0 + \sum_{j \in [m] \setminus \{k\}} \hat{\mu}_j \, A^j \, y_j \succeq 0,$$

which shows the claim. □

**Remark 6.19.** In the linear case with a linear inequality $a^\top y - a_0 \geq 0$, where $a \in \mathbb{R}^n_+$, $a_0 \in \mathbb{R}$, and the variables $y$ are binary, coefficient tightening would tighten coefficient $a_j$ to $\min\{a_j, a_0\}$. If $a_j > a_0 \geq 0$, then $\tilde{\mu}_j = a_0/a_j < 1$. Thus, Lemma 6.18 would change coefficient $a_j$ to $\tilde{\mu}_j \cdot a_j = a_0$, i.e., the same tightening. In this sense, Lemma 6.18 generalizes coefficient tightening from the linear case, see also Remark 6.12.

## 6.6 Computational Experiments

In this section, we empirically demonstrate the impact of the presented presolving routines for the SDP-based branch-and-bound approach and the LP-based cutting-plane approach.

We use SCIP-SDP 4.0 for solving the MISDPs, where all the routines mentioned in the previous sections are implemented. SCIP-SDP interfaces with SCIP 7.0.4 [114], and we use MOSEK 9.2.40 [181] for solving the continuous SDP relaxations in the SDP-based approach. The continuous LP relaxations in the cutting-plane approach are solved using SoPlex 5.0.2. All tests were performed on a Linux cluster with 3.5 GHz Intel Xeon E5-1620 Quad-Core CPUs with 32 GB main memory and 10 MB cache. The computations were run single-threaded and with a time limit of one hour.

### 6.6.1 Instances

We use a testset consisting of 185 instances for different applications, which are very briefly described in the subsequent paragraphs. Namely, 43 instances are Cardinality Least Squares (CLS) problems, 32 instances are Min-$k$-Partitioning (MkP) problems, 38 instances are Truss Topology Design (TTD) problems, and 46 instances are RIP problems. Moreover, there are 26 random MISDPs in the testset. These applications (with the exception of random MISDPs) are described in more detail in

**Table 6.2.** Overview over problem characteristics in the testset used for evaluating the presolving techniques.

| | | | variables | | SDP | | LP |
|---|---|---|---|---|---|---|---|
| application | # | $A^k \succeq 0$ | continuous | binary | #blocks | blocksizes | #constraints |
| (CLS) | 43 | 24 | 1 | 64 – 128 | 1 | 63 – 367 | 1 |
| (MkP) | 32 | 0 | 0 | 276 – 2415 | 1 | 24 – 70 | 24 – 70 |
| (RIP) | 46 | 0 | 465 – 2485 | 30 – 70 | 1 | 30 – 70 | 1802 – 9802 |
| (RND) | 26 | 13 | 60 – 120 | 60 – 120 | 1 | 60 – 120 | 0 |
| (TTD) | 38 | 38 | 0 | 27 – 384 | 1 | 16 – 44 | 0 – 127 |
| total | 185 | 75 | 0 – 2485 | 30 – 2415 | 1 | 16 – 367 | 0 – 9802 |

Gally [110]. Moreover, the RIP problem has already been treated in Definition 5.2. For 24 CLS problems, all 38 TTD problems, and 13 random MISDPs, all matrices $A^k$ are positive semidefinite. Thus, for these 75 instances, the two tightening procedures from Section 6.4 can be applied. Note that the random MISDPs and the RIP instances in our testset are larger than the random MISDPs and RIP instances used by Kobayashi and Takano [145]. Table 6.2 provides a short overview over the problem sizes of the instances within the testset, ordered by type. The first column lists the application and the second column displays the number of instances for the respective application. The third column presents the number of instances of the respective application for which all coefficient matrices $A^k \succeq 0$. The subsequent columns list the minimal and maximal number of continuous and binary variables, SDP-blocks, blocksizes of the SDP-blocks, and the number of LP constraints among all instances of the application, respectively. We remark that Table 6.2 lists the sizes of the instances after standard presolving of SCIP has been performed, e.g., linear constraints representing variable bounds have been eliminated and the bounds of the variables have been adapted accordingly. Most importantly, this did not include any of the MISDP presolving techniques introduced in this chapter.

**Cardinality Constrained Least Squares** Given is a data set with $d$ features, a matrix $A \in \mathbb{R}^{m \times d}$, whose rows represent sample points for the features, and a vector $b \in \mathbb{R}^m$ which contains the corresponding measurements. The goal in classical linear regression is to fit weights $x \in \mathbb{R}^d$ so that $Ax$ approximates $b$ best, i.e., $\|Ax - b\|_2$ is minimized. Since in many applications, only few of the features influence the measurements, a sparsity constraint on $x$ can be imposed. For a fixed sparsity level $k \in \mathbb{N}$, the (regularized) cardinality constrained least squares problem then reads

$$\inf_{x \in \mathbb{R}^d} \Big\{ \tfrac{1}{2} \|Ax - b\|_2^2 + \tfrac{1}{2} \rho \|x\|_2^2 \,:\, \|x\|_0 \leq k \Big\},$$

where $\frac{1}{2}\rho\|x\|_2^2$ is a regularization term for a given positive $\rho \in \mathbb{R}$. Pilanci et al. [201] showed that this problem is equivalent to the following MISDP:

$$\begin{aligned} \inf \quad & \tau \\ \text{s.t.} \quad & \begin{pmatrix} \mathbb{I}_m + \frac{1}{\rho} A \operatorname{Diag}(z)\, A^\top & b \\ b^\top & \tau \end{pmatrix} \succeq 0, \\ & \sum_{j=1}^{d} z_j \leq k,\ z \in \{0,1\}^d. \end{aligned} \tag{CLS}$$

We note that Bertsimas and Van Parys [23] present a very effective method to solve an equivalent convex formulation. We nevertheless add CLS instances to our testset, since they have distinctive features and complement the other problem types. We used a subset of the instances in [110], namely, 19 of the 20 instances based on real-world data and 24 of the 45 randomly generated instances. See [110, Chapter 3.5] for details on the generation of these instances, which are completely dense.

**Minimum $k$-Partitioning** Given is an undirected graph $G = (V, E)$ with $n$ nodes, edge-weights $c \colon E \mapsto \mathbb{R}$, and a positive integer $k \geq 2$. The minimum $k$-partitioning problem seeks to find a partitioning of $V := \{1, \dots, n\}$ into $k$ sets $V_1, \dots, V_k$ such that

$$\sum_{i=1}^{k} \sum_{e \in E[V_i]} c(e)$$

is minimized. We use an MISDP formulation that is based on Frieze and Jerrum [105], Eisenblätter [81, 82] and Ghaddar et al. [118]. Define the costs as $C_{ij} := c(\{i,j\})$ for $\{i,j\} \in E$ and $C_{ij} = 0$ otherwise. This leads to the formulation

$$\begin{aligned} \inf \quad & \sum_{1 \leq i < j \leq n} C_{ij}\, Y_{ij} \\ \text{s.t.} \quad & \tfrac{-1}{k-1}\, \mathbf{1}_n + \tfrac{k}{k-1}\, Y \succeq 0, \\ & Y_{ii} = 1,\ Y \succeq 0,\ Y \in \{0,1\}^{n \times n}, \end{aligned} \tag{MkP}$$

where $\mathbf{1}_n$ is the $n \times n$ all-one matrix. Additionally, using node weights $w \in \mathbb{R}^n$, we add the following constraint with lower and upper bounds $\ell$, $u$ on the weights of the parts:

$$\ell \leq \sum_{j=1}^{n} w_j\, Y_{ij} \leq u \qquad \forall\, i \in [n].$$

This ensures that the sum of weights of the nodes for each nonempty part in the partition lies in the interval $[\ell, u]$. We use 32 of the 59 instances in [110, Chapter 3.5], which all contain very sparse SDP constraints, since every $A^k$ consists of a single nonzero entry.

**Restricted Isometry Property** Recall from Section 5.2 that the (squared) lower and upper restricted isometry constants $\alpha_s^2$ and $\beta_s^2$ are defined as

$$\begin{aligned}\alpha_s^2 &:= \operatorname*{arg\,max}_{\alpha \geq 0} \{\alpha \|x\|_2^2 \leq \|Ax\|_2^2 \; \forall\, x \in \Sigma_s\} = \min\{\|Ax\|_2^2 \,:\, \|x\|_2^2 = 1,\, \|x\|_0 \leq s\},\\ \beta_s^2 &:= \operatorname*{arg\,min}_{\beta \geq 0} \{\beta \|x\|_2^2 \geq \|Ax\|_2^2 \; \forall\, x \in \Sigma_s\} = \max\{\|Ax\|_2^2 \,:\, \|x\|_2^2 = 1,\, \|x\|_0 \leq s\},\end{aligned}$$

respectively, see (5.25) and (5.26). These problems can be formulated as

$$\begin{aligned} \max / \min \quad & \langle A^\top A, X\rangle_{\mathrm{F}} \\ \text{s.t.} \quad & \operatorname{tr}(X) = 1, \\ & -z_j \leq X_{ij} \leq z_j \quad \text{for } i,\ j \in [n], \\ & \sum_{i=1}^n z_i \leq s, \\ & X \succeq 0, \quad z \in \{0,1\}^n, \end{aligned} \tag{RIP}$$

see (5.28). We use 46 instances which are created analogously to the instances in [111, Section 6]. Namely, the following six types of random matrices $A \in \mathbb{R}^{m \times n}$ are used:

| | |
|---|---|
| $0 \pm 1$ | $\mathbb{P}(A_{ij} = \sqrt{3/m}) = \mathbb{P}(A_{ij} = -\sqrt{3/m}) = \frac{1}{6}$ and $\mathbb{P}(A_{ij} = 0) = \frac{2}{3}$, |
| band | band matrix, entries uniformly in $\{0,1\}$, bandwidths 3, 5, 7, $m = n$, |
| Bernoulli | $A_{ij}$ uniformly in $\{\pm\sqrt{1/m}\}$, |
| binary | $A_{ij}$ uniformly in $\{0,1\}$, |
| normal | $A_{ij} \sim \mathcal{N}(0,1)$, |
| scaled normal | $A_{ij} \sim \mathcal{N}(0, \frac{1}{m})$. |

Here $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with parameters $\mu$ and $\sigma^2$. The sizes are given by $(m, n, k) \in \{(15, 30, 5), (25, 35, 4), (30, 40, 3), (40, 60, 5)\}$. The band matrix instances are larger with $(m, n, k) \in \{(40, 40, 3), (60, 60, 5), (70, 70, 4)\}$. For matrices of type $0 \pm 1$, Bernoulli and scaled normal, Baraniuk et al. [15] showed that for large $n$, sufficiently small $s$ and given $\delta$, these matrices satisfy the RIP of order $s$ and constant $\delta$ with high probability. As in Minimum $k$-Partitioning, the coefficient matrices $A^k$ only consist of one single nonzero entry, if (RIP) is written in form (6.1).

**Random MISDPs** We also consider random instances of the form

$$\sup \{b^\top y \,:\, \sum_{k=1}^{m} A^k\, y_k - A^0 \succeq 0,\ y \in \{0,1\}^{m_b} \times \mathbb{R}^{m_c}\}, \tag{RND}$$

where $m = m_b + m_c$ and $A^k \in \mathbb{R}^{n \times n}$ are symmetric matrices for $k \in [m]_0$. These instances are produced in the same way as done by Kobayashi and Takano [145]. More precisely, let $\mathcal{U}(C)$ denote the uniform distribution on the set $C$. Then, we choose a vector $y^*$ with $y^*_k \sim \mathcal{U}(\{0,1\})$ for $k \leq m_b$ and $y^*_k \sim \mathcal{U}([0,1])$ for $k > m_b$. Moreover, we choose the entries $A^k_{ij}$ of the coefficient matrices as $A^k_{ij} \sim \mathcal{U}([-1,1])$ for $k \in [m]$ and $1 \leq i \leq j \leq n$. In order to ensure that there exists a feasible solution to (RND), we set $A^0 = \sum_{k=1}^{m} A^k\, y^*_k - \alpha\mathbb{I}$, as well as $b_k = \langle A^k, \mathbb{I}\rangle_{\mathrm{F}}$ for $k \in [m]$, and $\alpha \geq 0$. For the definition of the inner product $\langle A^k, \mathbb{I}\rangle_{\mathrm{F}}$, see (1.1). Thus, the instances are generated based on the feasible solution $y^*$. For half of the instances, all coefficient matrices $A^k$ are ensured to be positive semidefinite and to have rank 1 by randomly choosing $a^k \sim \mathcal{U}([-1,1]^n)$ and setting $A^k = a^k(a^k)^\top$. The dimension of $A^k$ as well as the numbers of binary variables $m_b$ and continuous variables $m_c$ vary between $\{60, 90, 120\}$. The nonnegativity factor $\alpha$ is chosen as $\alpha \in \{0.1, 10\}$.

**Truss Topology Optimization** Truss topology optimization seeks truss structures that are stable with minimal total volume. Given is a ground structure, which is specified by a simple directed graph $D = (V, E)$ with $n$ nodes, $n_f$ of which are free, while the remaining nodes are fixed. The goal is to choose cross-sectional areas coming from a discrete set $\mathcal{A}$ for the bars on the edges. The model includes ellipsoidal robustness with respect to uncertain loads on the free nodes in $\{Qf \,:\, \|f\|_2 \leq 1\}$ for some matrix $Q$, following Ben-Tal and Nemirovski [21], and uses binary variables for choosing bars, see Mars [173]. This yields the model:

$$\begin{aligned}
\inf \quad & \sum_{e \in E} \ell_e \sum_{a \in \mathcal{A}} a\, x^a_e \\
\text{s.t.} \quad & \begin{pmatrix} 2\,\tau\,\mathbb{I} & Q^\top \\ Q & A(x) \end{pmatrix} \succeq 0, \\
& \sum_{a \in \mathcal{A}} x^a_e \leq 1 && \forall\, e \in E, \\
& \tau \leq C_{\max}, \\
& x^a_e \in \{0,1\} && \forall\, e \in E,\ a \in \mathcal{A}.
\end{aligned} \tag{TTD}$$

The binary variables $x^a_e$ choose a bar on edge $e$ with cross-sectional area $a \in \mathcal{A}$. The stiffness matrix $A(x)$ is given by $A(x) = \sum_{e \in E} \sum_{a \in \mathcal{A}} A_e\, a\, x^a_e$ with appropriate matrices $A_e$. The length of the bar on edge $e \in E$ is $\ell_e$ and $C_{\max}$ provides an upper bound on the compliance, which is the potential energy in the system. We use 38 of the 60 instances in [110, Chapter 3.5].

### 6.6.2 Settings

We use the following names for the algorithmic variants in which each different presolving routine described above is active and all other routines are deactivated.

- Basic linear inequalities:

| | |
|---|---|
| `DGZ` | add (DGZ) in presolving; |
| `DZI` | add (DZI) in presolving; |

- Tightening procedures only in presolving:

| | |
|---|---|
| `TM` | use Lemma 6.18 in presolving; |
| `TB-Pre` | apply Lemma 6.11 only in presolving; |

- Linear inequalities based on $2 \times 2$ minors:

| | |
|---|---|
| `2ML` | add (2ML) in presolving; |
| `2MP` | add (2MP) in presolving; |
| `2MV` | add (2MV) in presolving; |

- Propagation of variable bounds and tightening procedures:

| | |
|---|---|
| `PropUB-Pre` | apply (PropUB) only in presolving; |
| `PropUB` | apply (PropUB) every time propagation is called; |
| `PropTB` | apply Lemma 6.11 every time propagation is called. |

- Combinations of routines:

| | |
|---|---|
| `nopresol` | none; |
| `MIX1` | `DZI`, `TB-Pre`, `2MV`, `PropUB-Pre`, `PropUB`, `PropTB`; |
| `MIX2` | `DGZ`, `DZI`, `PropUB-Pre`, `PropUB`; |
| `allpresol` | `DGZ`, `DZI`, `TM`, `TB-Pre`, `2ML`, `2MP`, `2MV`, `PropUB-Pre`; |
| `allprop` | `PropUB`, `PropUB-Pre`, `PropTB`, `TB-Pre`; |
| `allprop-DGZ` | `DGZ`, `TB-Pre`, `PropUB`, `PropUB-Pre`, `PropTB`; |
| `allpresol-prop` | all routines activated in presolving and propagation. |

As outlined in the introduction of this chapter, presolving in SCIP and SCIP-SDP is applied in several rounds, so that different methods can influence each other. Moreover, since SCIP-SDP is based on SCIP, all presolving which SCIP applies by default, is also applied by SCIP-SDP. Thus every setting described above uses at least the standard presolving from SCIP for, e.g., linear constraints, and the listed additional presolving techniques, which are all based on the SDP constraints. Note that `MIX1` is the default setting for SCIP-SDP 4.0 when using the SDP-based approach. If there is no additional prefix, then the SDP-based approach is used for solving the MISDPs. The prefixes `LPA` and `LPE` denote that the LP-based cutting-plane approach is used instead of the SDP-based approach, in the following two variants: In LPA, eigenvector cuts are separated, and in LPE, eigenvector cuts are only enforced, see Section 6.1. For the settings MIX1-NoCA and LPA-MIX2-NoCA we additionally deactivated conflict analysis. Finally, in the setting CONC: MIX1 + LPA-MIX2 we used the concurrent mode of SCIP, where the instances are solved in parallel with settings MIX1 and LPA-MIX2, and solving stops, once the first setting reports an optimal solution. Note that our settings LPA-DGZ and LPE-DGZ roughly correspond to the branch-and-cut algorithm and the cutting-plane algorithm by Kobayashi and Takano [145], respectively.

### 6.6.3 Results for general MISDPs

Table 6.3 displays the results using the described testset for various settings listed in Section 6.6.2. Shown are the number of instances that were solved to optimality within the time limit of one hour out of all 185 instances (# opt) and the shifted geometric means of the number of nodes (# nodes) as well as the CPU time in seconds (time), see (1.2) for the definition of the shifted geometric mean. The next columns list the shifted geometric mean of the CPU time in seconds used for presolving (time), the arithmetic mean of the number of domain reductions (# reds), i.e., changed bounds, and added constraints (# addcons) in presolving for SDP constraints. The section "SDP constraints" in Table 6.3 shows the arithmetic means of the number of propagation calls (# prop), domain reductions (# reds), applied cuts (# cuts) and cutoffs (# cutoff) from SDP constraints. The last section "SDP timings" shows the shifted geometric means of the total time (total) and the propagation time (prop) spent for SDP constraints. For the shifted geometric means, we used a shift of $s = 100$ for nodes and $s = 1$ seconds for time, respectively. Tables 7.1 to 7.5, which can be found in Appendix B, present the results for each class of instances. Note that when comparing the number of used nodes for two settings in the following, we only take into account instances which have been solved to optimality by both settings, whereas the numbers in Table 6.3 and Tables 7.1 to 7.5 also take into account instances which ran into the time limit.

First of all, it turns out that Constraints (2MP) and coefficient tightening in Lemma 6.18 (TM), as well as bound tightening in Lemma 6.11 (TB-Pre) were never active in presolving throughout our testset. All other routines added constraints and/or changed bounds in presolving and produced domain reductions deeper within the branch-and-bound tree. In comparison to the setting nopresol in which all presolving routines are deactivated, adding the constraints (DGZ) or (2ML) has a negative effect on the running time, whereas adding the constraints (DZI) results in a speed-up of about 5 %. The latter is in line with the results reported by Mars [173] and Gally [110]. Using Lemma 6.8, i.e., adding (2MV) in presolving yields a minor improvement of the overall running time. Using Lemma 6.6 in propagation and/or

**Table 6.3.** Comparison of presolving routines using the SDP- and LP-based approach for all 185 MISDP instances.

| | | | | SDP presolving | | | SDP constraints | | | | SDP timings | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | time | #reds | #addcons | #prop | #reds | #cuts | #cutoff | total | prop |
| nopresol | 168 | 1405.3 | 180.23 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.25 | 0.08 |
| DGZ | 168 | 1395.9 | 187.41 | 0.00 | 11.0 | 13.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.25 | 0.08 |
| DZI | 168 | 1313.7 | 171.47 | 0.00 | 0.0 | 0.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.23 | 0.07 |
| TM | 168 | 1404.6 | 180.63 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.26 | 0.09 |
| TB-Pre | 167 | 1403.3 | 180.86 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.26 | 0.09 |
| 2ML | 168 | 1388.0 | 184.19 | 0.02 | 0.0 | 940.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.25 | 0.08 |
| 2MP | 168 | 1404.6 | 180.23 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.25 | 0.08 |
| 2MV | 167 | 1373.6 | 177.54 | 0.05 | 0.0 | 10 083.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.25 | 0.08 |
| PropUB-Pre | 168 | 1246.2 | 168.73 | 0.01 | 494.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.25 | 0.08 |
| PropUB | 168 | 1246.2 | 169.08 | 0.01 | 494.4 | 0.0 | 649 386.8 | 0.1 | 0.0 | 0.0 | 0.75 | 0.54 |
| PropTB | 167 | 1297.6 | 152.43 | 0.00 | 0.0 | 0.0 | 24 418.5 | 3378.6 | 0.0 | 330.5 | 1.62 | 1.57 |
| MIX1 | 167 | 1085.2 | 139.52 | 0.06 | 494.4 | 10 083.9 | 443 191.6 | 2211.1 | 0.0 | 148.2 | 2.73 | 2.69 |
| MIX1-NoCA | 167 | 1083.7 | 139.31 | 0.06 | 494.4 | 10 083.9 | 444 250.1 | 2224.3 | 0.0 | 149.3 | 2.68 | 2.64 |
| MIX2 | 168 | 1152.2 | 166.72 | 0.01 | 505.4 | 14.4 | 526 541.0 | 1365.6 | 0.0 | 0.0 | 0.70 | 0.51 |
| allpresol | 168 | 1176.8 | 168.12 | 0.08 | 505.4 | 11 038.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.22 | 0.07 |
| allprop | 166 | 1050.2 | 156.29 | 0.01 | 494.4 | 0.0 | 209 689.0 | 4135.1 | 0.0 | 2988.7 | 4.54 | 4.50 |
| allprop-DGZ | 166 | 1039.3 | 164.10 | 0.01 | 505.4 | 13.6 | 205 861.1 | 5427.9 | 0.0 | 2969.1 | 4.72 | 4.67 |
| allpresol-prop | 168 | 984.1 | 156.01 | 0.08 | 505.4 | 11 038.2 | 181 104.9 | 5612.6 | 0.0 | 1865.4 | 4.39 | 4.35 |
| LPA-nopresol | 104 | 386.5 | 346.38 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 46 014.9 | 108.7 | 4.24 | 0.01 |
| LPA-DGZ | 103 | 388.9 | 361.43 | 0.00 | 11.0 | 13.6 | 0.0 | 0.0 | 46 317.6 | 109.4 | 4.35 | 0.02 |
| LPA-DZI | 99 | 363.9 | 332.25 | 0.00 | 0.0 | 0.7 | 0.0 | 0.0 | 45 825.5 | 46.1 | 4.16 | 0.01 |
| LPA-TM | 104 | 387.0 | 347.25 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 45 977.5 | 108.7 | 4.28 | 0.02 |
| LPA-TB-Pre | 101 | 389.8 | 350.96 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 47 209.5 | 113.7 | 4.31 | 0.02 |
| LPA-2ML | 103 | 386.7 | 348.31 | 0.02 | 0.0 | 940.6 | 0.0 | 0.0 | 46 126.8 | 108.7 | 4.31 | 0.01 |
| LPA-2MP | 103 | 387.1 | 348.52 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 46 023.2 | 108.7 | 4.29 | 0.01 |
| LPA-2MV | 101 | 369.1 | 355.30 | 0.05 | 0.0 | 10 083.2 | 0.0 | 0.0 | 44 719.7 | 45.3 | 4.27 | 0.01 |
| LPA-PropUB-Pre | 82 | 381.3 | 347.90 | 0.01 | 494.4 | 0.0 | 0.0 | 0.0 | 46 073.0 | 108.7 | 4.26 | 0.01 |
| LPA-PropUB | 103 | 381.3 | 348.54 | 0.01 | 494.4 | 0.0 | 78 845.4 | 0.0 | 46 190.6 | 108.9 | 4.49 | 0.10 |
| LPA-PropTB | 103 | 372.0 | 469.86 | 0.00 | 0.0 | 0.0 | 15 194.2 | 4708.4 | 49 122.7 | 416.8 | 9.48 | 2.55 |
| LPA-MIX1 | 101 | 335.0 | 414.18 | 0.06 | 494.4 | 10 083.9 | 67 464.5 | 3490.7 | 42 739.7 | 152.6 | 8.18 | 2.13 |
| LPA-MIX2 | 98 | 360.4 | 345.39 | 0.01 | 505.4 | 14.4 | 77 337.6 | 0.0 | 46 421.8 | 47.2 | 4.52 | 0.10 |
| LPA-MIX2-NoCA | 102 | 352.7 | 329.60 | 0.01 | 505.4 | 14.4 | 64 138.4 | 0.0 | 41 606.0 | 52.7 | 4.37 | 0.09 |
| LPA-allpresol | 99 | 367.0 | 353.88 | 0.08 | 505.4 | 11 038.2 | 0.0 | 0.0 | 45 664.9 | 48.3 | 4.32 | 0.01 |
| LPA-allprop | 105 | 357.0 | 564.16 | 0.01 | 494.4 | 0.0 | 65 408.9 | 3194.6 | 39 697.9 | 907.7 | 15.34 | 4.11 |
| LPA-allpresol-prop | 99 | 327.8 | 641.33 | 0.08 | 505.4 | 11 038.2 | 57 647.1 | 4215.2 | 42 549.8 | 885.8 | 17.54 | 5.20 |
| LPE-MIX2 | 84 | 91 093.1 | 622.82 | 0.01 | 505.4 | 14.4 | 1 403 044.8 | 89 356.6 | 150 384.9 | 7665.8 | 22.71 | 2.36 |
| CONC: MIX1+LPA-MIX2 | 160 | 702.0 | 133.20 | 0.01 | 505.4 | 14.4 | 89.7 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 |

in presolving (PropUB, PropUB-Pre) also speeds up the solution process by 6 % and reduces the number of used nodes by 11 %. The highest impact of all routines alone is achieved by using bound tightening from Lemma 6.11 in propagation (PropTB), resulting in a 15 % reduction of the solution time. Interestingly, it solves one instance less than using no presolving at all. Using all presolving routines (allpresol) yields only a minor further improvement over the best pure presolving routine (DZI). If all propagation methods are activated as well (allpresol-prop), we obtain a major improvement in terms of overall running time (13 % faster) and processed nodes (28 % fewer nodes). Using only bound tightening and propagation (allprop) results in a further speed-up, and using the combination MIX1 turns out to be the best setting in terms of overall running times, which is about 22 % faster and processes about 23 % fewer nodes than using no presolving.

We also conducted experiments where the optimal objective value was set as objective limit and all primal heuristics are turned off in order to remove the impact of primal solutions. In this case, propagation via PropUB and PropTB reduces the number of nodes by 9 % and 10 %, respectively, compared to using no presolving or propagation (nopresol). Activating all propagation routines (allprop) results in a decrease of the number of nodes of 19 %. The propagation routines typically cut off nodes deeper in the tree. Thus, the speed-up of the solution process when using propagation routines can at least partly be explained by the fact that fewer nodes are needed to close the gap between the dual bound and the optimal (primal) objective value.

For all considered settings, the time spent for executing presolving or propagation is neglectable, so that all routines presented in this chapter can safely be activated without needing a significant amount of time by themselves. However, adding constraints or tightening bounds in presolving or deeper within the tree of course has effects on the solution process. Especially, primal heuristics are affected and may find primal solutions in a different order or not at all, which clearly influences the overall solution time.

In case of the LP-based cutting-plane approach, it turns out that DZI is the only setting which improves the running times (around 4 % faster), whereas 2MV and propagating the bound tightening (PropTB) have a negative impact. Moreover, only enforcing eigenvector cuts (LPE-MIX2) is clearly much worse than separating them (LPA-MIX2).

Concerning conflict analysis, it turns out that it has almost no impact when using the SDP-based approach, but it negatively influences the performance of the LP-based cutting-plane approach, regardless of the instance class. For the setting MIX2, deactivating conflict analysis results in a speed-up of almost 5 % for the solution time.

**Table 6.4.** Summary of the results for different presolving settings for each instance class separately.

**(a)** Results for the 43 Cardinality Constrained Least Squares (CLS) instances.

| | | | | SDP presolving | | SDP constraints | | |
|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #reds | #addcons | #prop | #reds | #cuts |
| `nopresol` | 41 | 382.5 | 201.05 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| `PropUB` | 41 | 382.6 | 200.65 | 0.0 | 0.0 | 36 176.0 | 0.0 | 0.0 |
| `PropTB` | 41 | 334.2 | 99.87 | 0.0 | 0.0 | 3813.9 | 6.9 | 0.0 |
| `LPA-nopresol` | 43 | 201.1 | 7.53 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 |
| `LPA-MIX2-NoCA` | 43 | 198.6 | 8.89 | 1.0 | 0.0 | 6009.3 | 0.0 | 2815.1 |

**(b)** Results for the 32 Minimum $k$-Partitioning (MkP) instances.

| | | | | SDP presolving | | SDP constraints | | |
|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #reds | #addcons | #prop | #reds | #cuts |
| `nopresol` | 32 | 181.5 | 63.23 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| `PropUB` | 32 | 181.5 | 64.14 | 0.0 | 0.0 | 209 081.2 | 0.0 | 0.0 |
| `PropTB` | 32 | 181.5 | 63.18 | 0.0 | 0.0 | 499.4 | 0.0 | 0.0 |
| `LPA-nopresol` | 5 | 67.3 | 2737.16 | 0.0 | 0.0 | 0.0 | 0.0 | 24 934.4 |
| `LPA-MIX2-NoCA` | 5 | 57.5 | 2408.40 | 0.0 | 0.0 | 24 964.9 | 0.0 | 22 153.9 |

**(c)** Results for the 46 Restricted Isometry Property (RIP) instances.

| | | | | SDP presolving | | SDP constraints | | |
|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #reds | #addcons | #prop | #reds | #cuts |
| `nopresol` | 36 | 4376.2 | 259.70 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| `PropUB` | 36 | 2756.7 | 199.05 | 1988.3 | 0.0 | 184 929.1 | 0.2 | 0.0 |
| `PropTB` | 36 | 4377.1 | 259.67 | 0.0 | 0.0 | 25 049.6 | 0.0 | 0.0 |
| `LPA-nopresol` | 0 | 35.7 | 3600.32 | 0.0 | 0.0 | 0.0 | 0.0 | 12 779.8 |
| `LPA-MIX2-NoCA` | 0 | 30.0 | 3600.70 | 2031.5 | 0.0 | 10 652.8 | 0.0 | 13 777.7 |
| `LPE-MIX2` | 33 | 41 373.9 | 366.98 | 2031.5 | 0.0 | 532 210.1 | 359 368.7 | 131 484.5 |

**(d)** Results for the 26 random MISDP (RND) instances.

| | | | | SDP presolving | | SDP constraints | | |
|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #reds | #addcons | #prop | #reds | #cuts |
| `nopresol` | 25 | 98.4 | 268.58 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| `PropUB` | 25 | 98.3 | 268.69 | 0.0 | 0.0 | 2903.5 | 0.0 | 0.0 |
| `PropTB` | 25 | 98.3 | 269.00 | 0.0 | 0.0 | 120.6 | 0.0 | 0.0 |
| `LPA-nopresol` | 26 | 99.8 | 413.63 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 |
| `LPA-MIX2-NoCA` | 25 | 98.6 | 418.66 | 0.0 | 96.9 | 1828.1 | 0.0 | 44 157.1 |

**(e)** Results for the 38 Truss Topology Design (TTD) instances.

| | | | | SDP presolving | | SDP constraints | | |
|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #reds | #addcons | #prop | #reds | #cuts |
| `nopresol` | 34 | 23 840.6 | 187.36 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| `DZI` | 34 | 17 547.3 | 146.73 | 0.0 | 3.6 | 0.0 | 0.0 | 0.0 |
| `PropUB` | 34 | 23 841.3 | 187.69 | 0.0 | 0.0 | 2 718 636.0 | 0.0 | 0.0 |
| `PropTB` | 33 | 18 695.5 | 182.75 | 0.0 | 0.0 | 83 737.6 | 16 440.6 | 0.0 |
| `LPA-nopresol` | 30 | 17 646.4 | 210.55 | 0.0 | 0.0 | 0.0 | 0.0 | 155 497.6 |
| `LPA-MIX2-NoCA` | 29 | 13 935.7 | 154.24 | 0.0 | 3.6 | 270 283.5 | 0.0 | 133 823.3 |

Table 6.4 provides a short extract of the results for each separate instance class and prominent settings. The full results are presented in Tables 7.1 to 7.5 in Appendix B. It turns out that for Min-$k$-Partitioning and random MISDPs, none of the routines has any impact on the performance, even if some constraints are added during presolving. No bounds are changed in presolving and no domain reductions are found deeper in the tree. For Cardinality Least Squares, using bound tightening from Lemma 6.11 in presolving and propagation (`PropTB`) reduces the overall running time by almost a factor of 2. Using bound tightening only in presolving (`TB-Pre`) or using the propagation from Lemma 6.6 in propagation and/or presolving (`PropUB`, `PropUB-Pre`) has almost no impact. For the RIP, the performance impact is switched. Using bound tightening (`PropTB`, `TB-Pre`) has no impact, whereas the propagation of Lemma 6.6 (`PropUB`, `PropUB-Pre`) significantly improves the performance; the solution process is about 23 % faster. Finally, for Truss Topology Design, Inequalities (DZI) turn out to be very effective and reduce the solution time by about 22 %, whereas bound tightening and propagation have no impact.

Interestingly, the winner between SDP- and LP-based approach also heavily depends on the instance class. Namely, for Cardinality Least Squares, the LP-based approach is faster by almost a factor 20, whereas for Min-$k$-Partitioning, the SDP-approach is almost a factor 35 times faster. For random MISDPs and Truss Topology Design, there is not much difference, but the SDP-approach is slightly faster. Lastly, for the RIP, the LP-based approach only solves a single instance within the time limit for the best setting, whereas the SDP-approach solves 36 out of 46. Moreover, the RIP instances are the only ones for which enforcing eigenvector cuts is significantly faster than separating eigenvector cuts. Using a concurrent solving mode with the best SDP-based setting `MIX1` and the best LP-based setting `LPA-MIX2` yields the best performance overall on the testset, resulting in 41 % fewer processed nodes and a solution process which is 26 % faster than using no presolving at all.

Overall, it turns out that several of the presented methods have a positive impact on the performance of SCIP-SDP, at almost no additional time spent for executing these methods. Most importantly, the inequalities (DZI) and (2MV) should be added during presolving, and the propagation in (PropUB) as well as the bound tightening from Lemma 6.11 (`TB`) should be executed both in presolving and in propagation calls deeper in the tree. Depending on the instance, it is beneficial to turn off one or more of these routines to gain improved performance, and to switch to an LP-based approach. By using the concurrent mode with an SDP and LP solving procedure run in parallel, one can exploit this performance difference between SDP- and LP-approach automatically.

### 6.6.4 Results for the RIP

In this section, we focus on the MISDP formulation of the RIP and evaluate the effect of the presolving methods from Chapter 6 and the special components presented in Section 5.3. In order to have a larger testset than the 46 RIP instances used for the general computational experiments in the previous section, we generated 180 new RIP instances, with the same six types of random matrices as described in the corresponding paragraph in Section 6.6.1. That is, we use $0 \pm 1$, band, Bernoulli, binary, normal and scaled normal matrices. There are three combinations for the size $m \times n$ of $A$ and the sparsity level $s$, namely

$$(m, n, k) \in \{(15, 30, 5),\ (25, 35, 4),\ (30, 40, 3)\},$$

and five instances for each type of randomness in the matrix $A$ and combination of $(m, n, k)$. Since the band matrices are square matrices, we use

$$(m, n, k) \in \{(30, 30, 5),\ (35, 35, 4),\ (40, 40, 3)\}$$

for these instances instead. Moreover, we compute the lower and upper RIC for each matrix, so that we obtain a testset of 180 instances overall. Compared to the 46 instances used in the last section, we omit the combination $(m, n, k) = (40, 60, 5)$ and also do not use larger band matrix instances.

As "default" formulation we use (5.28), i.e.,

$$\min_{X \succeq 0,\ z \in \{0,1\}^n} / \max \Big\{ \langle A^\top A, X \rangle_{\mathrm{F}} \,:\, \mathrm{tr}(X) = 1,\ \sum_{i=1}^{n} z_i \leq s,\ -z_j \leq X_{ij} \leq z_j\ \forall\, i, j \in [n] \Big\}.$$

We employ the same setup as in the previous section. Again, all computations were run single-threaded and with a time limit of one hour. All tables report the number of optimally solved instances within the time limit, and the shifted geometric means of the number of nodes and the solution times. We use a shift of $s = 1$ seconds and $s = 100$ nodes, see (1.2). In comparison to the results in the previous section, there is one notable difference. Since it turned out that conflict analysis has almost no impact on the SDP-based approach, and negatively influences the LP-based approach, we deactivated conflict analysis by default for all settings.

We use several of the settings introduced in Section 6.6.2. Furthermore, specific settings for the MISDP formulation of the RIP (5.28) are employed. In order to compare the influence of the bounds on the off-diagonal entries in $X$, we use the following three settings, which all include the basic bounds $0 \leq X_{ii} \leq z_i$ for all $i \in [n]$:

| | |
|---|---|
| `diagonal bounds` | no bounds for $X_{ij}$ with $i \neq j$; |
| `weak bounds` | $-z_j \leq X_{ij} \leq z_j$ for all $i$, $j \in [n]$ with $i \neq j$; |
| `strong bounds` | $-\frac{1}{2}z_j \leq X_{ij} \leq \frac{1}{2}z_j$ for all $i$, $j \in [n]$ with $i \neq j$, see (5.29). |

Besides, we use the following two additional settings for the LP-based approach:

| | |
|---|---|
| `LPE-SDP` | solve an additional SDP in enforcing if all integer variables are fixed; |
| `sparsify` | add multiple sparse eigenvector cuts as described in Section 5.3. |

If `LPE-SDP` is used, then eigenvector cuts are enforced as in the setting `LPE` and if all integer variables are already fixed, an additional SDP is solved. This ensures that solving the current node is finished without needing further eigenvector cuts. The setting `sparsify` enables the sparsification of the eigenvector cuts in separation or enforcing. In each execution of the sparsification routine, we allow to add at most 100 sparse eigenvector cuts. The target size for sparsification is set to 5, since this is the maximal sparsity level $s$ appearing in the instances in our testset. Thus, the produced eigenvector cuts are very sparse with only 5 nonzero entries. Finally, a `+` between different settings denotes their combination.

In Table 6.5 we compare several presolving and propagation routines for the SDP-based approach. The baseline setting is `nopresol` which does not activate any of the presolving and propagation routines described in this chapter. We omit the settings `DZI`, `2MP` and `TM`, since the results in the last section showed that these methods do not apply to the RIP instances. Furthermore, the settings `2MV` and `PropTB` do not have any impact on the RIP instances. It turns out that using `DGZ` and `2ML` has a negative impact on the solution time, but does not influence the number of nodes. Thus, the SDP relaxations become harder to solve due to the additional inequalities. This shows that adding the proposed linear inequalities, which are redundant in the

**Table 6.5.** Comparison of presolving and propagation routines for the SDP-based approach on the 180 RIP instances.

| setting | #opt | #nodes | time |
|---|---:|---:|---:|
| `nopresol` | 180 | 3524.5 | 79.92 |
| `DGZ` | 180 | 3534.7 | 95.90 |
| `2ML` | 180 | 3463.4 | 93.78 |
| `2MV` | 180 | 3524.5 | 80.85 |
| `PropUB` | 180 | 2211.6 | 59.27 |
| `PropTB` | 180 | 3524.5 | 79.91 |
| `MIX1` | 180 | 2222.5 | 60.34 |
| `strong bounds` | 180 | 2211.6 | 58.83 |

**Table 6.6.** Comparison of imposing different bounds on off-diagonal entries in the 180 RIP instances for the SDP-based approach.

| setting | #opt | #nodes | time |
|---|---|---|---|
| `diagonal bounds` | 121 | 2974.9 | 666.96 |
| `weak bounds` | 180 | 3524.5 | 79.92 |
| `strong bounds` | 180 | 2211.6 | 58.83 |
| `diagonal bounds + PropUB` | 180 | 3573.2 | 61.20 |
| `weak bounds + PropUB` | 180 | 2211.6 | 59.27 |
| `strong bounds + PropUB` | 180 | 2211.6 | 59.25 |

SDP relaxation, does not seem to help to approximate the SDP cone. The best single routine is `PropUB`, which significantly reduces both the number of nodes and the solution time. The combination `MIX1` of several presolving and propagation routines, which is the default setting of SCIP-SDP 4.0, performs equally good. This is due to the fact that `PropUB` strengthens the off-diagonal bounds to $-\frac{1}{2}z_j \leq X_{ij} \leq \frac{1}{2}z_j$, c.f. (5.29). If these stronger bounds are directly used in the problem formulation with the setting `strong bounds`, then no additional presolving or propagation is needed in order to obtain the same performance as the default formulation with `PropUB`. This also indicates that propagation mostly tightens only the off-diagonal bounds, if not already present in the problem formulation. Furthermore, using additional presolving routines on top of propagation does not have any impact.

Table 6.6 evaluates the effect of the bounds used in the problem formulation for the off-diagonal entries $X_{ij}$, $i \neq j$ on the SDP-based approach. We test the three bound versions `diagonal bounds`, `weak bounds` and `strong bounds` once without any additional presolving or propagation routine, and also with activated propagation in `PropUB`. Note that `weak bounds` is the default formulation. First of all, using no bounds on $X_{ij}$ with $i \neq j$ cannot solve all instances within the time limit and the solution time is at least 9 times slower than using either the weak or the strong bounds. Moreover, the smaller number of nodes in comparison to `weak bounds` indicates that solving the SDP relaxation takes much more time so that fewer nodes can be processed in the time limit. Using the stronger bounds is slightly better compared to the weak bounds in terms of solution time and needs significantly fewer nodes, which shows that a tighter SDP relaxation leads to an increased performance. These findings underline the importance of (strong) variable bounds in a branch-and-bound algorithm. Activating the propagation `PropUB` in the absence of bounds massively improves the performance by reducing the solution time by a factor of 10. By that, all instances could be solved within the time limit and the solution time is equal to the best setting `strong bounds` without propagation. But note that the number of nodes is significantly larger than using propagation with weak or strong

**Table 6.7.** Comparison of separation and enforcing eigenvector cuts for the LP-based approach on the 180 RIP instances.

| setting | #opt | #nodes | time |
|---|---|---|---|
| `LPA-MIX2 + 2ML` | 14 | 59.5 | 3052.76 |
| `LPA-MIX2 + 2ML + sparsify` | 105 | 2451.2 | 1194.54 |
| `LPE-MIX2 + 2ML` | 163 | 19 036.8 | 112.62 |
| `LPE-SDP-MIX2` + `2ML` | 165 | 19 248.2 | 116.12 |
| `LPE-MIX2 + sparsify + 2ML` | 166 | 18 464.2 | 112.72 |

bounds. This may be due to the fact that the estimate in the root node is too weak and much more nodes are needed in order to obtain an SDP relaxation which is as tight as in the presence of bounds in the problem formulation. Altogether, the results show that propagation has a significant impact if the problem formulation is loose, and tight bounds are one key point to obtain a good performance in a branch-and-bound scheme.

Let us now switch from the SDP-based approach to the LP-based approach. Recall that in SCIP-SDP, eigenvector cuts can either be enforced (`LPE`) or separated (`LPA`), see Section 6.1. Moreover, it is possible to solve an additional SDP in enforcing, if all integer variables are fixed (`LPE-SDP`). Table 6.7 compares these three possibilities, and the effect of sparsifying the eigenvector cuts in separation and enforcing. We always use the combination `MIX2`, which is the default setting of the LP-based approach in SCIP-SDP. Additionally, we add the inequalities (2ML), since they have a very positive effect for the RIP instances, as we will see when comparing presolving and propagation for the LP-based approach.

First of all, it turns out that separation only solves very few instances within the time limit. The very small number of processed nodes already indicates that it takes very long to solve the LP relaxation in the nodes. Thus, it seems that many eigenvector cuts are needed in order to obtain a feasible solution of the LP relaxation which satisfies the SDP constraint. By that, the LP relaxation becomes increasingly larger and harder to solve. Sparsifying the eigenvector cuts and adding multiple sparse eigenvector cuts greatly improves the performance. More than half of the instances can now be solved, and the solution time decreases by almost a factor of 3. The increased number of nodes indicates that the LP relaxations are now easier to solve, most likely due to the sparsity of the added cuts. Moreover, since multiple cuts are added in one separation round, it seems that fewer rounds are needed, which also decreases the solution time. Another reason for the performance difference may be that the dense eigenvector cuts added in separation can lead to numerical instabilities in the LP relaxation in comparison to the sparse eigenvector cuts.

**Table 6.8.** Comparison of presolving, propagation and the formulation of the bounds for off-diagonal entries for the LP-based approach on the 180 RIP instances.

| setting | #opt | #nodes | time |
|---|---|---|---|
| LPE-nopresol | 150 | 37 886.0 | 269.16 |
| LPE-DGZ | 156 | 35 875.7 | 187.47 |
| LPE-PropUB | 162 | 27 995.2 | 169.38 |
| LPE-PropTB | 150 | 37 887.3 | 269.56 |
| LPE-2ML | 164 | 29 702.3 | 170.13 |
| LPE-2MV | 151 | 37 607.4 | 279.15 |
| LPE-nopresol + strong bounds | 162 | 27 859.9 | 175.11 |
| LPE-MIX2 + 2ML + diagonal bounds | 148 | 19 894.3 | 1473.55 |
| LPE-MIX2 + 2ML + weak bounds | 163 | 19 036.8 | 112.62 |
| LPE-MIX2 + 2ML + strong bounds | 164 | 18 928.5 | 112.49 |

If the eigenvector cuts are not separated but enforced, this massively improves the performance. Most of the instances can be solved and the solution time is decreased by one order of magnitude. Moreover, the number of nodes is increased by almost one order of magnitude as well. This shows that it is much faster to first solve the LP relaxation to optimality and then enforce the SDP constraint by eigenvector cuts. Besides, the individual nodes seem to be solved significantly faster so that much more nodes can be processed. Solving an additional SDP in enforcing seems to have a very slight negative influence. Using multiple sparse eigenvector cuts in enforcing slightly reduces the number of nodes. Most importantly, it solves three additional instances.

The effect of presolving, propagation and the used bounds for $X_{ij}$, $i \neq j$ in the problem formulation is investigated for the LP-based approach in Table 6.8. We use the approach of enforcing eigenvector cuts (LPE), since Table 6.7 showed that this is the fastest of the different variants. The baseline setting is LPE-nopresol, which uses no other presolving or propagation routine presented in this chapter. As in Table 6.5, we omit the settings DZI, 2MP and TM, since they do not apply to the MISDP formulation of the RIP. First of all, it turns out that in contrast to the SDP-based approach, the approximation of the SDP cone by using DGZ, 2ML improves the performance. Both settings solve more instances, and lead to a faster solution time. Using 2MV seems to slightly increase the solution time but solves one more instance compared to LPE-nopresol. The approximation by DGZ decreases the solution time significantly, and decreases the number of nodes slightly. In addition, six more instances can be solved. The greatest impact of these inequalities is achieved by 2ML, which solves 14 more instances, uses only two thirds of the nodes and also reduces the time by about 30 % compared to LPE-nopresol. This already shows that in contrast to the SDP-based approach, solving LP relaxations greatly profits from adding additional linear inequalities which approximate the SDP cone. This is due to the fact that these inequalities are not redundant in the LP relaxation.

Using the propagation in `PropUB` is slightly faster and uses slightly fewer nodes than `2ML`, but solves two instances less. Moreover, adding the stronger bounds to the problem formulation but using no additional technique in `LPE-nopresol + strong bounds` is slightly worse compared to `PropUB`. For both the formulation with weak bounds and strong bounds, activating further presolving and propagation with the combination `LPE-MIX2 + 2ML` has a significant impact, in comparison to `LPE-nopresol` and `LPE-nopresol + strong bounds`, respectively. Using no bounds on the off-diagonal entries $X_{ij}$, $i \neq j$ significantly deteriorates the performance, even if the propagation by Inequalities (PropUB) is used in the setting `LPE-MIX2 + 2ML + diagonal bounds`. The solution time increases by one order of magnitude, and 15 instances less can be solved. However, the number of nodes does not increase significantly. This indicates again, that the LP relaxation becomes much harder to solve, which results in an increased total solution time. Overall, the best setting `LPE-MIX2 + 2ML + strong bounds` solves 14 instances more, uses only half the number of nodes and decreases the solution time by about 59 % compared to using no presolving or propagation and the weak bounds (`LPE-nopresol`). These results indicate that for the LP-based approach, various presolving and propagation routines need to be activated in order to achieve the best performance. Moreover, it also shows that the routines influence each other and thus lead to further strengthenings. However, in contrast to the SDP-based approach, bounds on the off-diagonal entries are strictly necessary even if the propagation `PropUB` is used.

In Table 6.9 we compare the LP-based and the SDP-based approach. We use the SDP setting `PropUB` and the LP setting `LPE-MIX2 + 2ML + weak bounds`, which are one of the best settings for the respective approach. Since the results of the comparison depend on the size of the instance and whether the lower RIC $\alpha_s^2$ or the upper RIC $\beta_s^2$ is computed, we divide the instances into large ($n = 40$, $s = 3$), medium ($n = 35$, $s = 4$) and small ($n = 30$, $s = 5$) ones. Note that if $n$ and thus the blocksize of the SDP constraint decreases, the sparsity level $s$ increases. Table 6.9 shows the results for each size, divided into lower and upper RIC (30 instances each), and both RICs together (60 instances). Moreover, the table also presents the results when all 180 instances are taken into consideration, and if all lower or upper RICs are solved (90 instances each). As a first observation, the upper RIC seems to be overall easier to solve than the lower RIC. For each size, both the LP- and the SDP-based approach are faster and use fewer nodes when solving the upper RIC compared to the lower RIC, even if the difference for the large instances and the SDP-based approach is only very small. Taking all instances into consideration confirms that the upper RIC is easier than the lower RIC, since even the slower

**Table 6.9.** Comparison of the LP- and the SDP-based approach on the 180 RIP instances, separately for the small, medium and large instances, as well as divided into lower and upper RIC.

**(a)** Large instances with $n = 40$ and $s = 3$.

| | lower RIC (30) | | | upper RIC (30) | | | both RICs (60) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| `PropUB` | 30 | 1928.8 | 59.90 | 30 | 1292.0 | 58.98 | 60 | 1580.5 | 59.44 |
| `LPE-MIX2 + 2ML` | 30 | 10 687.1 | 53.05 | 30 | 3751.3 | 37.31 | 60 | 6345.5 | 44.51 |

**(b)** Medium instances with $n = 35$ and $s = 4$.

| | lower RIC (30) | | | upper RIC (30) | | | both RICs (60) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| `PropUB` | 30 | 3637.7 | 83.51 | 30 | 1728.9 | 58.93 | 60 | 2514.6 | 70.17 |
| `LPE-MIX2 + 2ML` | 30 | 50 273.7 | 176.71 | 30 | 11 537.5 | 64.72 | 60 | 24 112.0 | 107.07 |

**(c)** Small instances with $n = 30$ and $s = 5$.

| | lower RIC (30) | | | upper RIC (30) | | | both RICs (60) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| `PropUB` | 30 | 5676.5 | 76.58 | 30 | 1268.4 | 32.38 | 60 | 2711.5 | 49.89 |
| `LPE-MIX2 + 2ML` | 13 | 127 537.4 | 1193.06 | 30 | 15 700.5 | 73.50 | 43 | 44 808.1 | 297.25 |

**(d)** All instances.

| | lower RIC (90) | | | upper RIC (90) | | | both RICs (180) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| `PropUB` | 90 | 3425.1 | 72.64 | 90 | 1415.9 | 48.33 | 180 | 2211.6 | 59.27 |
| `LPE-MIX2 + 2ML` | 73 | 40 986.2 | 224.52 | 90 | 8813.4 | 56.24 | 163 | 19 036.8 | 112.62 |

LP-based approach for the upper RIC is better than the faster SDP-based approach for the lower RIC. Moreover, it is interesting to see that the LP-based approach struggles much more with the lower RIC than the SDP-based approach. For the small instances, it cannot even solve all 30 instances within the time limit.

Let us consider the lower RIC $\alpha_s^2$ in more detail. If the size decreases and the sparsity level increases, the SDP-based approach needs increasingly more nodes. The solution time, however, first increases when comparing large to medium instances, but then decreases again when comparing medium to small instances. This indicates that the blocksize of the SDP constraint seems to be more important than the sparsity level for the difficulty to solve the instance. In contrast, the LP-based approach suffers from a dramatic increase in solution time and used nodes for decreasing size and increasing sparsity level. Going from large to medium instances, the number of nodes increases by almost a factor of 5, and solution time increases by roughly a factor of 2.5. Comparing medium to small instances, the number of nodes increases again by a factor of 2.5 and the solution time increases by a factor of about 5.6. Additionally, only about half of the small instances for the lower RIC can be solved by the LP-based approach. This demonstrates that for the LP-based approach, the sparsity level seems to be much more important for the ability to solve instances. One reason is that the LP relaxation scales much better to higher dimensions, i.e., larger values of the blocksize $n$ of the SDP constraint. In contrast, scalability is still a problem when solving SDPs. For more information, we refer to the recent survey by Majumdar et al. [170] and the references therein.

For the upper RIC $\beta_s^2$ as well as increasing sparsity level $s$ and decreasing size $n$, the SDP-based approach becomes faster. Between large and medium instances, there is almost no difference in the solution time, but the number of nodes increases slightly. Going from medium to small instances, the solution time decreases by almost a factor of 2, and also the number of nodes decreases again. This underlines once more that the blocksize $n$ is more important than the sparsity level $s$ when using the SDP-based approach. For the LP-based approach, we can draw the same conclusion as for the lower RIC $\alpha_s^2$. However, the performance loss is nowhere as dramatic as for the lower RIC.

For the large instances, it turns out that the LP-based approach is faster for each RIC separately and also for both RICs combined. This changes for the medium and small instances, where the SDP-based approach becomes clearly faster, again for each RIC separately and the combination of both RICs.

Overall, the upper RIC is considerably easier to solve, regardless of whether the SDP- or the LP-based approach is used. For large instances with a very small sparsity level, the LP-based approach outperforms the SDP-based approach, but whenever the sparsity level is increased, or the blocksize is decreased, the SDP-based approach is better.

These results can be confirmed by considering instances with increased parameters $(m, n, s)$. Therefore, we created 180 RIP instances completely analogous to the ones used above with sizes

$$(m, n, s) \in \{(40, 60, 5),\ (50, 70, 4),\ (60, 80, 3)\},$$

and

$$(m, n, s) \in \{(60, 60, 5),\ (70, 70, 4),\ (80, 80, 3)\}$$

for the band matrices. Table 6.10 shows the results for these larger instances when using again the settings `PropUB` and `LPE-MIX2 + 2ML`. First of all, note that only very few instances can be solved within the time limit. The increasing difficulty to solve the instances depending on the blocksize $n$ and the sparsity level $s$ can

**Table 6.10.** Comparison of the LP- and the SDP-based approach on the 180 larger RIP instances, separately for the small, medium and large instances, as well as divided into lower and upper RIC.

**(a)** Large instances with $n = 80$ and $s = 3$.

| | lower RIC (30) | | | upper RIC (30) | | | both RICs (60) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| PropUB | 5 | 953.7 | 2255.78 | 5 | 1184.2 | 2886.18 | 10 | 1063.3 | 2551.59 |
| LPE-MIX2 + 2ML | 25 | 89 511.0 | 2153.06 | 25 | 41 631.9 | 1497.95 | 50 | 61 052.6 | 1795.90 |

**(b)** Medium instances with $n = 70$ and $s = 4$.

| | lower RIC (30) | | | upper RIC (30) | | | both RICs (60) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| PropUB | 5 | 1365.0 | 1817.11 | 5 | 1260.1 | 2145.58 | 10 | 1311.6 | 1974.53 |
| LPE-MIX2 + 2ML | 5 | 62 228.4 | 2863.74 | 5 | 31 396.9 | 1408.77 | 10 | 44 207.5 | 2008.63 |

**(c)** Small instances with $n = 60$ and $s = 5$.

| | lower RIC (30) | | | upper RIC (30) | | | both RICs (60) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| PropUB | 5 | 3166.5 | 1261.41 | 5 | 1927.5 | 1590.90 | 10 | 2473.5 | 1416.61 |
| LPE-MIX2 + 2ML | 2 | 58 915.4 | 3344.34 | 5 | 25 230.4 | 1299.88 | 7 | 38 563.7 | 2085.11 |

**(d)** All instances.

| | lower RIC (90) | | | upper RIC (90) | | | both RICs (180) | | |
|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | #opt | #nodes | time | #opt | #nodes | time |
| PropUB | 15 | 1614.8 | 1729.23 | 15 | 1424.2 | 2143.76 | 30 | 1516.7 | 1925.37 |
| LPE-MIX2 + 2ML | 32 | 68 977.7 | 2742.22 | 35 | 32 070.6 | 1399.85 | 67 | 47 041.0 | 1959.31 |

be observed even clearer for the instances with overall increased blocksizes. For the lower RIC, the SDP-based approach becomes significantly faster if the blocksize decreases, even if the sparsity level is increased. The behavior of the LP-based approach is the direct opposite, its performance considerably deteriorates when the blocksize decreases but the sparsity level increases: In the worst case only 2 out of 30 instances can be solved. For the upper RIC, both the LP-based and SDP-based approach become faster when the blocksize is decreased and the sparsity level is increased. Moreover, the LP-based approach is always faster than the SDP-based approach, by almost a factor of 2 for $(n, s) = (80, 3)$, by a factor of 1.5 for $(n, s) = (70, 4)$ and still about 18 % faster for $(n, s) = (60, 5)$. Overall, these results show that the upper RIC is comparably easier to compute for both the LP- and the SDP-based approach. Moreover, for the instances with overall increased blocksize, it turns out that for the upper RIC, the LP-based approach is faster by almost 35%, whereas for the lower RIC, it is outperformed by the SDP-based

**Table 6.11.** Comparison of the effect of the nonnegativity constraint on the LP- and the SDP-based approach on the 45 binary RIP instances.

| setting | #opt | #nodes | time |
|---|---|---|---|
| `LPE-nopresol` | 32 | 73 698.7 | 440.51 |
| `LPE-nopresol` + $X_{ij} \geq 0$ | 45 | 77 216.5 | 169.97 |
| `LPE-MIX2 + 2ML` | 45 | 38 345.9 | 157.48 |
| `LPE-MIX2 + 2ML` + $X_{ij} \geq 0$ | 45 | 40 598.5 | 135.78 |
| `nopresol` | 45 | 34 052.9 | 328.91 |
| `nopresol` + $X_{ij} \geq 0$ | 45 | 34 002.2 | 298.44 |
| `MIX1` | 45 | 26 765.7 | 282.34 |
| `MIX1` + $X_{ij} \geq 0$ | 45 | 26 717.8 | 270.22 |

approach, which is almost 40 % faster. Additionally, even if only the blocksize was slightly increased and the sparsity level remained the same, the instances become considerably harder to solve. Only a fraction of the 180 instances (30 and 67) could be solved within the time limit of one hour with one of the approaches.

Lastly, we consider the effect of exploiting the possible nonnegativity of $A$ (componentwise) as in Lemma 5.7. Since Lemma 5.7 only applies to the maximization problem, nonnegativity constraints on the variables can only be added for computing the upper RIC $\beta_s^2$, and only if $A \geq 0$. Moreover, only two types of random matrices satisfy $A \geq 0$ componentwise, namely, band matrices and binary matrices. Since the instances generated with the band matrices are very easy to solve for the given parameters $(m, n, s)$, we test the nonnegativity only on binary matrices. To have a larger testset, we generate 45 new random binary matrices with the same parameters as before, i.e., 15 of each combination $(m, n, s) \in \{(15, 30, 5),\ (25, 35, 4),\ (30, 40, 3)\}$. In order to demonstrate the impact of the nonnegativity constraint, we use no further presolving or propagation technique. For a comparison, we also add the nonnegativity constraint to the LP and SDP settings `LPE-MIX2 + 2ML` and `MIX1`, respectively. The results are displayed in Table 6.11. It can be seen that regardless of using additional presolving and propagation or not, the nonnegativity constraint has a positive impact. Clearly, the effect is again most visible, when there is no additional presolving. Then, the LP-based approach (`LPE-nopresol`) can only solve 32 of the 45 instances within the time limit. Adding $X_{ij} \geq 0$ leads to a decrease of the solution time by about 59 %, whereas the number of nodes increases slightly. Most importantly, all 45 instances can now be solved. In the presence of additional presolving (`LPE-MIX2 + 2ML`), adding $X_{ij} \geq 0$ again results in slightly more processed nodes, but the solution time decreases by about 14 %. For the SDP-based approach, using the nonnegativity constraint decreases the solution time by about 9 % and 5 %, depending on whether additional presolving is used or not. Since the number of nodes remains almost the same in both cases, the SDP relaxation is again seemingly easier to solve. The results again demonstrate the impact of additional structure on the performance of the solution process.

We also conducted tests for the valid inequality $\sum_{i \neq j} X_{ij} \leq s-1$ from Lemma 5.4. However, it turned out that adding this inequality does not help in the solution process. Depending on the other techniques used, it either has no impact on the performance, or it even worsens the performance. Thus, it seems that this single inequality is not strong enough to improve the performance over the other presolving methods. Either more valid inequalities or a stronger inequality is needed in order to have a positive effect on the solution process.

## 6.7 Concluding Remarks and Outlook

In this chapter, we extended several presolving methods from mixed-integer linear programs to MISDPs and introduced new methods. On our testset, these methods are effective on average with a decrease of about 22 % in running time compared to using no presolving, when applied in the nodes, i.e., propagation is performed in the whole tree, see the results in Section 6.6.3. The impact, however, depends on the type of the instance. In the extreme, for partitioning instances presolving has no impact at all. For others, (node) presolving implies a performance improvement of about 25 % (RIP) or even 44 % (CLS), although in the latter case solving LPs is even better with an improvement of at least one order of magnitude between SDP and LP solving. These numbers illustrate again that the effectiveness of presolving depends on the type of application. However, since executing these methods only cause a negligible runtime increase, they can easily be used or tested on new instance types to see their effect on the solution process. This is true, in particular, if more instances are generated by modeling software in the future. Such instances can be expected to not be tuned as well as instances generated by humans, i.e., they may contain loose initial variable bounds or superfluous information in form of constraints. Consequently, presolving can have a large impact on these instances by providing tighter bounds and deriving effective valid inequalities, which overall tightens the formulation of the instance. Thus, the results of this chapter lead to the following conclusion: "*The presolving methods are effective if they can be applied; and if not, they only impose a very small overhead.*"

An open question for future research is to derive effective presolving based on larger minors of the positive semidefinite matrices $A(y)$. For larger minors, inequalities similar to (2ML) may however be less sparse, at least if used for a constraint $X \succeq 0$ in primal form, see (6.3) and (6.4). Furthermore, it would be interesting to investigate whether it is possible to predict the performance of presolving methods and whether switching to LP solving is advisable based on the application. The results specifically for the RIP showed that at least for the larger instances, the LP-based approach is better suited for computing the upper RIC, in contrast to the lower RIC.

Another interesting point is the application of presolving techniques developed for general SDPs, such as facial reduction to reduce the dimension of the SDP, or exploiting sparsity structure.

A detailed analysis of the presolving methods in this chapter and the special components from Section 5.3 for the MISDP formulation of the RIP on a larger testset of RIP instances reveals that the SDP-based approach highly benefits from strong bounds on the off-diagonal elements. These can either be added to the problem formulation, or be found by the propagation from Lemma 6.6 within the solution process. If the propagation is not used and no bounds are imposed on the off-diagonal elements in the problem formulation, then the performance deteriorates significantly. The additional inequalities (DGZ) and (2ML) do not have any positive impact on the performance when using the SDP-based approach, most likely since they are already implied by the SDP constraint and do not strengthen the problem formulation further. In contrast, for the LP-based approach, these additional inequalities improve the performance, and, again, the strongest possible bounds should be used.

Furthermore, it turned out that the upper RIC is much easier to compute than the lower RIC. Besides, the choice between the SDP-based and the LP-based approach depends on whether the upper or the lower RIC shall be computed. For the lower RIC, the SDP-based approach performs better, whereas the upper RIC seems to favor the LP-based approach. An analysis of this behavior and an answer whether this depends on the randomness in the matrix $A$ is an interesting open question. In Section 5.2 we have seen that the upper RIC is also known as sparse principal component analysis (SPCA), see (5.26). For the SPCA problem, there is an extensive amount of literature, where different authors have proposed various valid inequalities and other components for solving different formulations of this problem, see, e.g., Bertsimas et al. [24], d'Aspremont et al. [60], Dey et al. [65], and Li and Xie [157]. Investigating their impact on solving the MISDP formulation of the upper RIC, and using Lemma 5.8 to also exploit them for the MISDP formulation of the lower RIC is another interesting research direction for future work. The transformation of an instance of the minimization problem to an instance of the maximization problem may be helpful in its own regard, since the computational experiments suggested that the MISDP formulation of the upper RIC may be easier to solve.

Of course, more work needs to be done to experiment further with the sparsification of the eigenvector cuts. The experiments conducted for this thesis already showed that exploiting sparsity can have a significant impact, but no exhaustive studies with different sparsity levels and number of added cuts have been executed so far.

Besides, branching is currently only applied on integer variables with a fractional relaxation solution value, as described in Section 6.1. Thus, it may be promising to think about different branching strategies. One idea is to apply (spatial) branching on diagonal entries $X_{ii}$ of the matrix variable $X \in \mathcal{S}^n$. Since $0 \leq X_{ii} \leq 1$, we have $X_{ii} > \frac{1}{2}$ for at most one $i \in [n]$. Moreover, since at most $s$ diagonal entries $X_{ii}$ are nonzero, where $s$ is typically much smaller than the dimension $n$, most of the diagonal entries are zero. Thus, we can branch on the diagonal entries being greater or less than $\frac{1}{2}$. To do so, $n+1$ branching nodes need to be created. In the $i$-th node, set $X_{ii} \geq \frac{1}{2}$ and $X_{jj} \leq \frac{1}{2}$ for all $j \neq i$. The $(n+1)$-th node then has $X_{ii} \leq \frac{1}{2}$ for all $i \in [n]$. This branching step can also be applied recursively.

# CHAPTER 7

# Conclusion and Outlook

In this thesis we have presented a general framework for sparse recovery in the presence of side constraints. The proposed framework builds upon an already existing one by Juditsky et al. [137], and extends it by also incorporating additional side constraints. This is achieved by introducing a set $\mathcal{C}$ and adding the constraint $x \in \mathcal{C}$ to the recovery program, which enables to model additional knowledge available on the elements that are to be reconstructed. By that, it can be used to analyze the effect of exploiting structure in the recovery problem, which was one of the main research topics of the "EXPRESS" project within the SPP 1798. In Section 2.2, we have derived the general null space property (NSP$^{\mathcal{C}}$), which under some assumptions characterizes the ability to successfully reconstruct every sufficiently sparse element from its measurements under a linear measurement operator. These assumptions state conditions, which need to be satisfied in a specific setting in order to obtain a characterization of uniform recovery using the presented null space property. We have shown that several specific settings already treated in the literature, including cases with additional side constraints such as nonnegativity or positive semidefiniteness fit into our framework. Moreover, we have demonstrated that for these settings, the null space property (NSP$^{\mathcal{C}}$) simplifies to the respective null space properties already known in the literature (c.f. Section 1.1 and Example 2.12), which shows the generality of the proposed general null space property. Furthermore, in Section 2.3, the framework has been extended to cover robust recovery in the presence of noise. This includes stable recovery if the original signal is not exactly sparse. For both cases, we have proposed a slightly strengthened null space property which allows to control the reconstruction error in uniform recovery. Lastly, we have considered individual recovery, that is, the recovery of a fixed sparse element in Section 2.4.

Again, we were able to obtain known results for stable and robust recovery as well as individual recovery in specific settings as special cases.

In Chapter 3, we have considered three interesting side constraints in more detail. First, we have introduced a block-structure on matrices in Section 3.1, which generalizes the case of block-structured vectors. We have derived this setting with and without an additional positive semidefiniteness constraint from our general framework and presented the corresponding null space properties for characterizing uniform recovery. Moreover, we have compared the resulting null space properties and, for the special case of block-structured vectors, presented a family of measurement matrices which satisfy the NSP for block-sparse nonnegative vectors, but violate the NSP for general block-sparse vectors. This served as a first demonstration that exploiting additional side constraints can yield weaker recovery conditions, which are satisfied by more measurement matrices. Section 3.2 has treated the recovery of sparse integral vectors and highlighted differences between general sparse vectors and sparse integral vectors. Even if the corresponding side constraint $x \in \mathbb{Z}^n$ is nonconvex, this setting also fits into our framework and we were again able to derive the known null space properties for sparse integral vectors with and without additional variable bounds. Finally, we have considered constant modulus constraints in Section 3.3. Such constraints demand that the absolute value, or modulus, of each entry of a vector be constant, e.g., 0 or 1. This side constraint is especially important for complex vectors, and it frequently appears in signal processing applications, such as the problem of joint antenna selection and phase-only beamforming, which was considered as one example for additional structure in $x$. Analogously to the previous special cases, we have derived the constant modulus setting from our general framework and have introduced a corresponding null space property, which was not known in the literature before. Since the corresponding recovery problem is nonconvex, we also have presented a specialized solution algorithm to solve the recovery problem. We have used a general spatial branch-and-bound algorithm and added specific components to handle constant modulus constraints as well as a heuristic to obtain good solutions. Numerical results for joint-antenna selection and phase-only beamforming have shown the performance of the introduced components compared to using a standard spatial branch-and-bound algorithm.

In Chapter 4, we have investigated the null space property for sparse nonnegative vectors under Gaussian random measurement matrices. We have derived a lower bound for the minimal number of measurements needed for uniform recovery of sparse nonnegative vectors with high probability in Section 4.2. This was achieved by showing that the corresponding null space property is satisfied with high probability. The derived bound is non-asymptotic, whereas in the literature, only asymptotic bounds were previously known. In order to compute the bound, we have extended the known approach used for sparse vectors to sparse nonnegative vectors. Unfortunately, the obtained bound turned out to be weaker than the corresponding known bound for sparse vectors. However, simulations for the quantities involved in obtaining the bound have revealed that fewer measurements seem to be needed for uniform recovery if an additional nonnegativity constraint is present and exploited in the recovery process. This is underlined by a numerical comparison of individual recovery with and without nonnegativity. Thus, in theory, it should also be possible to derive a bound for sparse nonnegative vectors which is indeed smaller than the bound for sparse vectors. Furthermore, Section 4.3 has treated the case of block-structured matrices and also presented a lower bound on the minimal number of measurements needed for uniform recovery. For this setting, we have extended another proof technique for sparse vectors to block-structured matrices. The obtained results show that a random measurement operator allows for uniform recovery of block-sparse matrices if the number of measurements satisfies a lower bound which depends on the sparsity level $s$, the number of blocks $k$ and the block sizes $d_1 \cdot d_2$. Most importantly, this obtained lower bound scales logarithmically in $k$ and linearly in $d_1 \cdot d_2$, i.e., in the dimension of the single blocks, but it does not directly scale in the overall dimension $k \cdot d_1 \cdot d_2$. Thus, the block-structure is represented in the bound.

As a last part in our consideration of sparse recovery under side constraints, we have considered possibilities to verify recovery conditions in Chapter 5. In Section 5.1, we have presented several MIP formulations for checking whether a given measurement matrix satisfies null space properties for some specific settings. More precisely, testing the NSPs for sparse vectors, sparse nonnegative vectors and block-sparse vectors as well as block-sparse nonnegative vectors has been formulated as a MIP. A short numerical comparison has shown again that exploiting nonnegativity yields a null space property which is easier to verify for a given measurement matrix. Moreover, for a small dimension, we also have demonstrated numerically that the NSP for sparse nonnegative vectors is satisfied with high probability for fewer measurements than the NSP for sparse vectors. Afterwards, we have considered the RIP as another recovery condition for uniform recovery of sparse vectors in Sections 5.2 and 5.3. We have presented the well-known MISDP formulation of the RIP and shortly have discussed some properties of this formulation.

This has led us to consider presolving for general MISDPs in Chapter 6. Presolving is one of the most important steps in solving general mathematical optimization problems, and in contrast to MIPs, only few presolving techniques for MISDPs were known in the literature. Thus, we have introduced several new presolving methods for general MISDPs in Sections 6.2 to 6.5. Some of these methods are direct extensions of the respective methods for MIPs, whereas others were completely new.

An exhaustive numerical comparison in Section 6.6 has shown the effectiveness of the proposed methods for several classes of MISDPs. As one class of MISDPs, we have also paid special attention to the MISDP formulation of the RIP, and evaluated several methods for this formulation in detail. The results have shown that using the correct solution approach and activating some of the proposed presolving methods has a major impact on the performance.

## Outlook

Even if the thesis treated null space properties for sparse recovery under various aspects, there are still several open questions left. First of all, it would be natural to ask whether it is also possible to formulate a general RIP in our proposed general framework in Chapter 2. A general RIP in a slightly different general framework is presented by Traonmilin and Gribonval [238], and it would be interesting to compare the two frameworks in terms of possible additional side constraints and obtained recovery conditions. Another framework for sparse recovery, which generalizes many special cases known in the literature is the atomic setting in Chandrasekaran et al. [49]. The atomic setting assumes that an element is built as a linear combination of few elements taken from a so-called atomic set. Using the atomic norm, which is the gauge function of the convex hull of the set of atoms, recovery of sparse elements is possible. Here, sparse refers to elements whose linear combination only contains few nonzero coefficients. Clearly, taking the usual basis vectors of $\mathbb{R}^n$ as atomic set, we obtain the classical setting of sparse vectors with only few nonzero entries and the atomic norm is exactly the $\ell_1$-norm. For this setting, a simple optimality condition for individual recovery is presented in [49], as well as a lower bound on the minimal number of random Gaussian measurements needed for individual recovery with high probability. It is not directly clear that this atomic setting also fits into, or is comparable with our framework. Indeed, if the atomic set allows for non-unique representations of elements, then expressing sparsity through projections becomes nontrivial. Nevertheless, formulating a corresponding atomic null space property which characterizes uniform recovery and extending the lower bound also to uniform recovery would certainly be interesting.

Closely connected is the concept of decomposable norms considered by Negahban et al. [188], Candès and Recht [44], and Roulet et al. [214]. Individual recovery is treated in [44], where again an optimality condition is used to guarantee individual recovery. This optimality condition is then analyzed under random Gaussian measurements. Since we showed in Lemma 2.7 that our framework is connected with the concept of decomposable norms, it is natural to ask which results directly carry over to individual recovery in our framework as well, and especially if it is also possible to analyze individual recovery under random measurements. In [214], uniform recovery is treated and a null space property for the setting of decomposable norms is presented. It is reasonable to believe that this NSP also emerges from our general framework if we adopt the viewpoint of decomposable norms, see Lemma 2.7.

As outlined in more detail in Section 3.4, there are several settings with specific structure which have been considered in the literature. For each of those settings, adapted recovery conditions as well as explicit recovery algorithms are presented. Since the proposed recovery conditions for these settings are mostly an adaption of the RIP, it remains open to find an NSP for these cases. Such an NSP can be obtained from our framework, given that the setting fits into the framework and can be shown to satisfy the assumptions needed for our uniform recovery results. Obtaining an NSP instead of an RIP as recovery condition is especially important, since the NSP is a characterization, whereas the RIP is only a sufficient condition. Moreover, recently it has been shown by Dirksen et al. [66] that the RIP fails to capture cases in which successful recovery is possible and yields suboptimal bounds for the number of random measurements needed for uniform recovery of sparse vectors. This indicates that the NSP is better suited for obtaining precise statements about successful recovery.

Concerning the analysis of sparse recovery under random measurements it remains open to strengthen the bound on the minimal number of measurements needed for uniform recovery of sparse nonnegative vectors derived in Section 4.2. It turns out that this bound is worse than the bound for sparse vectors, but experiments and also empirical results for the quantity which needs to be bounded — the Gaussian width of the set of unit-norm vectors violating the NSP — show that there should be a gap between the bounds. For the case of sparse vectors, the Gaussian distribution could be used to derive the corresponding bounds. Since for sparse nonnegative vectors, the sign plays an important role in the recovery conditions, rectified Gaussian random vectors appear in the derivation of the bound in Theorem 4.7, see Appendix 7. Thus, any improvement of this bound will most likely involve more precise estimations for rectified Gaussian random vectors. Moreover, it remains open to derive such a bound in the case of block-sparse nonnegative vectors and block-sparse positive semidefinite matrices.

On top of that, for the recovery of low-rank matrices with and without additional positive semidefiniteness as well as block-structured matrices, it is an open question to formulate the problem to test whether a given measurement operator satisfies the corresponding NSP as optimization problem. Due to the possible positive semidefiniteness constraint, this will most likely not be a MIP formulation, but rather an MISDP formulation. The integrality is expected to enter as well, since for the NSP, we need to split the set of singular values, or eigenvalues of a matrix. Moreover, it would be interesting to investigate the obtained MIP formulations for the NSPs in case of (block-) sparse (nonnegative) vectors in more detail. Most certainly it is possible to speed-up the solution process by incorporating problem specific components.

Besides these computational aspects, an important question not treated within this thesis is the complexity of verifying an NSP condition. For the classical NSP, it is known that checking whether a given matrix satisfies the NSP is $\mathcal{NP}$-hard, see Tillmann and Pfetsch [237]. It would certainly be interesting to see whether the same also holds in the presence of additional side constraints. For example, the problem of recovering sparse and sparse nonnegative vectors are directly connected by using a variable split. To be more precise, the problem

$$\min \left\{ \|x\|_1 \,:\, Ax = b,\ x \in \mathbb{R}^n \right\} \tag{7.1}$$

is equivalent to the problem

$$\min \left\{ \left\| \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix} \right\|_1 \,:\, (A, -A) \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix} = b,\ x^{(1)},\, x^{(2)} \geq 0 \right\}, \tag{7.2}$$

since an optimal solution $x^*$ of (7.1) and an optimal solution $(x^{(1)})^*$, $(x^{(2)})^*$ of (7.2) are connected through $(x^{(1)})^* = (x^*)^+$ as well as $(x^{(2)})^* = (x^*)^-$. Consequently, we can either try to recover sparse vectors directly via (7.1) or try to recover their positive and negative part via (7.2). For the latter problem, we can invoke the NSP for the recovery of sparse nonnegative vectors. Accordingly, it may be possible to reduce the decision problem whether a given measurement matrix satisfies the classical NSP to the corresponding decision problem for the nonnegative NSP. This would show that the $\mathcal{NP}$-hardness result of testing the classical NSP also holds for the nonnegative NSP.

A further important aspect which was not covered throughout this thesis is a comparison of exploiting and disregarding the block-structure for block-sparse vectors. Clearly, it is possible to use ordinary $\ell_1$-minimization for recovering block-sparse vectors. If there are $k$ blocks, then every block-$s$-sparse vector $x$ is also $\tilde{s}$-sparse in the classical sense, where $\tilde{s}$ is the sum of the $s$ largest block sizes of the $k$ blocks, since $x$ has at most $\tilde{s}$ nonzero elements. The converse is however wrong in general, since not every sparse vector is also block-sparse with respect to some block-structure. Thus, the conditions for uniform recovery of non-block-sparse vectors may be too strong for uniform recovery of all block-sparse vectors. For a short discussion in terms of the restricted isometry constant and property, and an illustrative example, see Eldar and Mishali [89]. In case of additional nonnegativity, Theorem 3.14 shows that there exist matrices which satisfy the block-nonnegative NSP and violate the nonnegative NSP. Hence, the nonnegative NSP is indeed too strong for uniform recovery of block-sparse nonnegative vectors. Finding a similar example for the case of (block-) sparse vectors remains an open question.

Furthermore, in the block-structured settings, a mixed norm was used in the recovery problems. This mixed norm consists of applying some norm to each block and then using the $\ell_1$-norm (or, simply the sum) of the resulting numbers. Typically, the $\ell_2$-norm or the Frobenius norm are used on the blocks due to their robustness. The proposed null space properties derived in Section 3.1 for block-structured vectors and matrices without nonnegativity or positive semidefiniteness hold for arbitrary norms on the blocks, see Remark 3.7 and Corollary 3.9. In the presence of additional nonnegativity or positive semidefiniteness constraints, the corresponding null space properties explicitly use the $\ell_1$-norm or the nuclear norm on the blocks, see Theorem 3.4 and Corollary 3.8. Consequently, an interesting line of research would be to analyze the usage of different norms on the blocks in the case of block-sparse nonnegative vectors or block-sparse positive semidefinite matrices. It is important to notice that for non-block settings, replacing the $\ell_1$ norm in ordinary $\ell_1$-minimization (7.1) has negative side effects. For example, it is well known that for $q > 1$, recovery using the $\ell_q$-norm and the recovery problem

$$\min\{\|x\|_q \,:\, Ax = b\} \tag{7.3}$$

already fails for 1-sparse vectors, see Figure 1.1 for a simple example. If $0 < q < 1$, then using the $\ell_q$-norm has favorable recovery properties as shown by Mourad and Reilly [182], but the resulting recovery problem (7.3) is known to be $\mathcal{NP}$-hard, see Ge et al. [116]. However, note that replacing the $\ell_1$-norm by the $\ell_q$-norm with $0 < q < 1$ in the classical NSP yields a condition which characterizes uniform recovery of sparse vectors using (7.3), see Foucart and Rauhut [104], whereas $\ell_q$-minimization does not fit into our framework, since Assumption (A4) is violated as we have seen in Remark 2.17.

# Appendix

## A Bounds for Recovery of Sparse Nonnegative Vectors Under Random Measurements

In this section, we prove Theorem 4.7, which presents a lower bound on the minimal number of measurements needed for uniform recovery of sparse nonnegative vectors. The proof includes a detailed derivation of the bounds in (4.10) and (4.11).

Recall that a bound for the minimal number of measurements needed for uniform recovery can be obtained by Gordon's Escape Theorem 4.4. Therefore, we need to estimate the Gaussian width $\omega(T_s)$ of the set $T_s$ of unit-norm vectors violating (NSP$_{\geq 0}$). By using the convex cone $K_s$ defined as

$$K_s \coloneqq \{v \in \mathbb{R}^n \,:\, v_{s+1}, \dots, v_n \leq 0,\ \mathbb{1}^\top v \geq 0\},$$

as well as conic duality, we can estimate the Gaussian width as

$$\omega(T_s) = \mathbb{E}\Big[\min_{t>0,\ z_{s+1},\dots,z_n \leq t} \Big\{ \Big( \sum_{i=1}^{s} (\tilde{g}_i + t)^2 \Big)^{1/2} + \Big( \sum_{i=s+1}^{n} (\tilde{g}_i + z_i)^2 \Big)^{1/2} \Big\}\Big],$$

see Lemma 4.6. Consider a fixed $t > 0$, and let

$$E_1^{(\mathrm{nng})} \coloneqq \mathbb{E}\Big[\Big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2\Big)^{1/2}\Big], \quad E_2^{(\mathrm{nng})} \coloneqq \mathbb{E}\Big[\min_{z_{s+1},\dots,z_n \leq t} \Big(\sum_{i=s+1}^{n} (\tilde{g}_i + z_i)^2\Big)^{1/2}\Big].$$

In order to derive the bounds for $E_1^{(\mathrm{nng})}$ and $E_2^{(\mathrm{nng})}$ in (4.10) and (4.11), respectively, we first need to prove some auxiliary results. Throughout this section, we denote with $\varphi$ and $\Phi$ the pdf and cdf of the standard Gaussian distribution, respectively, i.e.,

$$\varphi(t) = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{t^2}{2}\Big), \qquad \Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} \exp\Big(-\frac{x^2}{2}\Big)\, dx.$$

We will need a result about the expectation of applying the exponential function to a scaled squared standard Gaussian random variable $X$, a function, which is also known as moment generating function of $X$.

**Lemma 7.1** (Foucart and Rauhut [104, Lemma 7.6]). *Let $X$ be a standard Gaussian random variable and let $\theta \in \mathbb{R}$ with $\theta < \frac{1}{2}$. Then,*

$$\mathbb{E}[\exp(\theta X^2)] = \frac{1}{\sqrt{1 - 2\theta}}.$$

A useful inequality for the expectation of general random vectors under convex functions is Jensen's inequality in the next lemma.

**Lemma 7.2** (Jensen's inequality, see, e.g., Foucart and Rauhut [104, Theorem 7.10]). *Let $X \in \mathbb{R}^n$ be a random vector and let $f \colon \mathbb{R}^n \mapsto \mathbb{R}$ be a convex function. Then we have $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$.*

We need the expectation of the maximal squared $\ell_2$-norm of multiple standard Gaussian random vectors in the next lemma.

**Lemma 7.3** (Foucart and Rauhut [104, Proposition 8.2]). *Let $g^{(1)}, \ldots, g^{(m)} \in \mathbb{R}^n$ be a collection of (not necessarily independent) standard Gaussian random vectors. For any $\kappa > 0$, we have the bound*

$$\mathbb{E}\big[\max_{i \in [m]} \|g^{(i)}\|_2^2\big] \leq (2 + 2\kappa) \ln(m) + n(1 + \kappa) \ln\big(1 + \tfrac{1}{\kappa}\big),$$

*and thus, for $\kappa = \sqrt{n/(2\ln(m))}$,*

$$\mathbb{E}\big[\max_{i \in [m]} \|g^{(i)}\|_2^2\big] \leq \big(\sqrt{2\ln(m)} + \sqrt{n}\big)^2.$$

Another important distribution that will be used in this chapter is the rectified standard Gaussian distribution. It is obtained from the standard Gaussian distribution by setting negative elements to 0. Its pdf is given by

$$\phi(t) = \frac{1}{2}\delta(t) + \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{t^2}{2}\Big) U(t),$$

where $\delta(x)$ is the Dirac delta distribution with $\delta(x) = 0$ for all $x \in \mathbb{R} \setminus \{0\}$, and $\delta(0) = \infty$, and $U(x)$ is the unit step function with $U(x) = 0$ for $x \leq 0$ and $U(x) = 1$ for $x > 0$. We use the notation $X \sim \mathcal{N}^R(0,1)$ to denote that $X$ is a rectified standard Gaussian random variable. If $X \sim \mathcal{N}(0,1)$, then $Y = \max\{X, 0\} \sim \mathcal{N}^R(0,1)$. In the following, unless noted otherwise, $h$ will denote a rectified standard Gaussian random vector of appropriate dimension.

The first two moments of a rectified standard Gaussian variable can now be computed as follows, see also Beauchamp [18].

**Fact 7.4.** *Let $X \sim \mathcal{N}(0,1)$ and let $Y := \max\{0, X\}$ be a rectified standard Gaussian random variable. Then, by the law of total expectation,*

$$\begin{aligned}\mathbb{E}[Y] &= \mathbb{P}(X > 0) \cdot \mathbb{E}[X|X > 0] + 0 \cdot \mathbb{P}(X \leq 0) = \frac{1}{2} \cdot \frac{2}{\sqrt{2\pi}} = \frac{1}{\sqrt{2\pi}},\\ \mathbb{E}[Y^2] &= \mathbb{P}(X > 0) \cdot \mathbb{E}[X^2|X > 0] + 0 \cdot \mathbb{P}(X \leq 0) = \frac{1}{2} \cdot 1 = \frac{1}{2},\end{aligned}$$

*since $\mathbb{E}[X|X > 0]$ and $\mathbb{E}[X^2|X > 0]$ are the first two moments of a truncated standard Gaussian random variable, which can be found, e.g., in Horrace [131].*

Using these results, the moment generating function of a rectified standard Gaussian random variable can be computed as follows.

**Lemma 7.5.** *Let $X \sim \mathcal{N}(0,1)$, let $Y := \max\{0, X\}$ be a rectified standard Gaussian random variable, and let $\theta \in \mathbb{R}$ with $\theta < \frac{1}{2}$. Then,*

$$\mathbb{E}\big[\exp(\theta\, Y^2)\big] = \frac{1}{2}\Big(1 + \frac{1}{\sqrt{1-2\theta}}\Big).$$

*Proof.* By the law of total expectation and Lemma 7.1,

$$\begin{aligned}\mathbb{E}\big[\exp(\theta\, Y^2)\big] &= \mathbb{E}\big[\exp(\theta\, X^2)\big] \cdot \mathbb{P}(X > 0) + \exp(\theta \cdot 0) \cdot \mathbb{P}(X \leq 0)\\ &= \frac{1}{2}\frac{1}{\sqrt{1-2\theta}} + \frac{1}{2},\end{aligned}$$

where we used Lemma 7.1 for the moment generating function $\mathbb{E}[\exp(\theta\, X^2)]$ of $X$. This finishes the proof. □

Lemma 7.5 yields the following estimate for the expectation of the maximum squared $\ell_2$-norm of a set of rectified standard Gaussian random vectors.

**Lemma 7.6.** *Let $h^{(1)}, \ldots, h^{(m)} \in \mathbb{R}^n$ be rectified standard Gaussian random vectors. Then, for any $\kappa > 0$,*

$$\mathbb{E}\big[\max_{i\in[m]} \|h^{(i)}\|_2^2\big] \leq (2 + 2\kappa)\Big[\ln(m) + n\ln\big(\tfrac{1}{2}\big) + n\ln\big(1 + \sqrt{1 + \tfrac{1}{\kappa}}\big)\Big].$$

*Proof.* Let $\theta \in \mathbb{R}$ with $\theta < \frac{1}{2}$. Since the logarithm is a concave function, we can use Jensen's inequality in Lemma 7.2 to obtain

$$\begin{aligned}\mathbb{E}\big[\max_{i\in[m]} \|h^{(i)}\|_2^2\big] &= \frac{1}{\theta}\mathbb{E}\Big[\ln\big(\max_{i\in[m]} \exp(\theta\|h^{(i)}\|_2^2)\big)\Big] \\ &\leq \frac{1}{\theta}\ln\Big(\mathbb{E}\big[\max_{i\in[m]} \exp(\theta\|h^{(i)}\|_2^2)\big]\Big) \\ &\leq \frac{1}{\theta}\ln\Big(\mathbb{E}\big[\sum_{i=1}^{m} \exp(\theta\|h^{(i)}\|_2^2)\big]\Big) \\ &\leq \frac{1}{\theta}\ln\Big(m\cdot\mathbb{E}\big[\exp(\theta\|h\|_2^2)\big]\Big) \\ &= \frac{1}{\theta}\ln\Big(m\cdot\prod_{j=1}^{n}\mathbb{E}\big[\exp(\theta h_j^2)\big]\Big),\end{aligned}$$

where $h$ is a rectified standard Gaussian random vector. The last equality is due to the independence of the entries of $h$. Lemma 7.5 yields

$$\begin{aligned}\mathbb{E}\big[\max_{i\in[m]} \|h^{(i)}\|_2^2\big] &\leq \frac{1}{\theta}\ln\Big(m\cdot\frac{1}{2^n}\Big(1+\frac{1}{\sqrt{1-2\theta}}\Big)^n\Big) \\ &= \frac{1}{\theta}\Big[\ln(m) + n\cdot\ln\Big(\frac{1}{2}\Big(1+\frac{1}{\sqrt{1-2\theta}}\Big)\Big)\Big].\end{aligned}$$

Since $\kappa > 0$ and $\theta < \frac{1}{2}$ we set $\theta = (2+2\kappa)^{-1} < \frac{1}{2}$, which yields

$$\mathbb{E}\big[\max_{i\in[m]} \|h^{(i)}\|_2^2\big] \leq (2+2\kappa)\Big[\ln(m) + n\ln\big(\tfrac{1}{2}\big) + n\ln\Big(1+\sqrt{1+\tfrac{1}{\kappa}}\Big)\Big].$$

This finishes the proof. □

The minimum of the $\ell_2$-norms of a set of rectified standard Gaussian random vectors $h^{(1)},\dots,h^{(m)} \in \mathbb{R}^n$ is considerably easier to estimate. Since the $h^{(i)}$ as well as their components are independent, we can estimate the expectation as

$$\mathbb{E}\big[\min_{i\in[m]} \|h^{(i)}\|_2^2\big] \leq \mathbb{E}\big[\|h^{(1)}\|_2^2\big] = n\mathbb{E}\big[Y^2\big] = \tfrac{1}{2}n, \tag{7.4}$$

where $Y$ is a rectified standard Gaussian random variable. The last equality is due to Fact 7.4. We are now ready to derive the bounds for $E_1^{(\mathrm{nng})}$ and $E_2^{(\mathrm{nng})}$ in (4.10) and (4.10), respectively. Recall that for a fixed $t \geq 0$,

$$E_1^{(\mathrm{nng})} := \mathbb{E}\Big[\Big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2\Big)^{1/2}\Big], \quad E_2^{(\mathrm{nng})} := \mathbb{E}\Big[\min_{z_{s+1},\dots,z_n\leq t}\Big(\sum_{i=s+1}^{n}(\tilde{g}_i + z_i)^2\Big)^{1/2}\Big].$$

**Estimating $\boldsymbol{E}_1^{(\mathsf{nng})}$** We define the vectors $h^+$, $h^- \in \mathbb{R}^n$ by $h_i^+ = \max\{0, g_i\}$ as well as $h_i^- = \max\{0, -g_i\}$ for $i \in [n]$. Then, $E_1^{(\mathrm{nng})}$ can be estimated as

$$\begin{aligned}
E_1^{(\mathrm{nng})} &= \mathbb{E}\Big[\Big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2\Big)^{1/2}\Big] \\
&\leq t\sqrt{s} + \mathbb{E}\Big[\Big(\sum_{i=1}^{s}\tilde{g}_i^2\Big)^{1/2}\Big] \\
&\leq t\sqrt{s} + \mathbb{E}\Big[\sqrt{\max_{|S|=s,\, S\subseteq[n]} \|h_S^+\|_2^2 + \min_{|T|=s,\, T\subseteq[n]} \|h_T^-\|_2^2}\,\Big].
\end{aligned}$$

Since $h_i^+$ and $h_i^-$ are rectified standard Gaussian random vectors, we can use Lemma 7.6 and (7.4) to estimate

$$\begin{aligned}
&\max_{|S|=s,\, S\subseteq[n]} \|h_S^+\|_2^2 \leq (2+2\kappa)\Big[\ln\tbinom{n}{s} + n\ln\big(\tfrac{1}{2}\big) + n\ln\big(1+\sqrt{1+\tfrac{1}{\kappa}}\big)\Big], \\
&\min_{|T|=s,\, T\subseteq[n]} \|h_T^-\|_2^2 \leq \tfrac{1}{2}s.
\end{aligned}$$

For the first inequality, we used that there are $\binom{n}{s}$ subsets of cardinality $s$, so that the maximum is taken over $\binom{n}{s}$ random vectors. Thus, we obtain the following estimate of $E_1^{(\mathrm{nng})}$:

$$\begin{aligned}
E_1^{(\mathrm{nng})} &= \mathbb{E}\Big[\Big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2\Big)^{1/2}\Big] \\
&\leq t\sqrt{s} + \mathbb{E}\Big[\Big(\sum_{i=1}^{s}\tilde{g}_i^2\Big)^{1/2}\Big] \\
&\leq t\sqrt{s} + \mathbb{E}\Big[\sqrt{\max_{|S|=s,\, S\subseteq[n]} \|h_S^+\|_2^2 + \min_{|T|=s,\, T\subseteq[n]} \|h_T^-\|_2^2}\,\Big] \\
&\leq t\sqrt{s} + \sqrt{\mathbb{E}\Big[\max_{|S|=s,\, S\subseteq[n]} \|h_S^+\|_2^2 + \min_{|T|=s,\, T\subseteq[n]} \|h_T^-\|_2^2\Big]} \\
&\leq t\sqrt{s} + \sqrt{\min_{\kappa>0}\Big\{(2+2\kappa)\Big[\ln\tbinom{n}{s} + s\ln\big(\tfrac{1}{2}\big) + s\ln\big(1+\sqrt{1+\tfrac{1}{\kappa}}\big)\Big]\Big\} + \tfrac{1}{2}s} \\
&\leq t\sqrt{s} + \sqrt{\min_{\kappa>0}\Big\{(2+2\kappa)\Big[s\ln\big(\tfrac{en}{s}\big) + s\ln\big(\tfrac{1}{2}\big) + s\ln\big(1+\sqrt{1+\tfrac{1}{\kappa}}\big)\Big]\Big\} + \tfrac{1}{2}s}.
\end{aligned}$$

Here, we used Jensen's inequality in Lemma 7.2 for the third inequality, since the square root function is concave. The last inequality uses the estimate $\binom{n}{s} \leq (\frac{en}{s})^s$, where $e = \exp(1)$, see, e.g., Foucart and Rauhut [104, Lemma C.5].

**Estimating $E_2^{(\mathrm{nng})}$** The remaining term, $E_2^{(\mathrm{nng})}$, can be estimated as follows:

$$\begin{aligned} E_2^{(\mathrm{nng})} &= \mathbb{E}\Big[\min_{z_{s+1},\ldots,z_n\leq t}\Big(\sum_{i=s+1}^{n}(\tilde{g}_i+z_i)^2\Big)^{1/2}\Big] \\ &\leq \Big(\mathbb{E}\big[\min_{z_i\leq t}\sum_{i=s+1}^{n}(\tilde{g}_i+z_i)^2\big]\Big)^{1/2} \\ &= \Big(\mathbb{E}\big[\sum_{i=s+1}^{n}\big(\min\{0,\tilde{g}_i+t\}\big)^2\big]\Big)^{1/2} \\ &\leq \Big((n-s)\cdot\mathbb{E}\big[\big(\min\{0,\tilde{g}_n+t\}\big)^2\big]\Big)^{1/2}. \end{aligned} \tag{7.5}$$

The inequality in (7.5) is due to $\tilde{g}_1 \geq \cdots \geq \tilde{g}_n$, so that

$$\big(\min\{0,\tilde{g}_1+t\}\big)^2 \leq \cdots \leq \big(\min\{0,\tilde{g}_n+t\}\big)^2,$$

since $\min\{0,\tilde{g}_i+t\} \leq 0$ for all $i \in [n]$. The probability that the smallest element $\tilde{g}_n$ of $n$ i.i.d. standard Gaussian random variables $g_1,\ldots,g_n$ is greater or equal than $-t$ can be computed as

$$\mathbb{P}(\tilde{g}_n \geq -t) = \mathbb{P}(g_1 \geq -t,\ldots,g_n \geq -t) = \prod_{i=1}^{n} 1-\mathbb{P}(g_i \leq -t).$$

Define the random variable $X := \big(\min\{0,\, \tilde{g}_n+t\}\big)^2$, and let $g$ be a standard Gaussian random variable. Since $\min\{0,\, \tilde{g}_n+t\} \leq 0$, the cdf $F_X(z)$ of $X$ for $z \geq 0$ is given by

$$\begin{aligned} F_X(z) = \mathbb{P}(X \leq z) &= \mathbb{P}\big(-\sqrt{z} \leq \min\{0,\, \tilde{g}_n+t\} \leq \sqrt{z}\big) \\ &= \mathbb{P}\big(\min\{0,\, \tilde{g}_n+t\} \leq \sqrt{z}\big) - \mathbb{P}\big(\min\{0,\, \tilde{g}_n+t\} < -\sqrt{z}\big) \\ &= 1-\mathbb{P}\big(\tilde{g}_n+t < -\sqrt{z}\big) = 1-\mathbb{P}\big(\tilde{g}_n < -\sqrt{z}-t\big), \end{aligned}$$

and $F(z) = 0$ for $z < 0$. For $z \geq 0$, we can further compute

$$\begin{aligned} F_X(z) = 1-\mathbb{P}\big(\tilde{g}_n < -\sqrt{z}-t\big) &= 1-\Big[1-\mathbb{P}\big(g_i \geq -\sqrt{z}-t\ \forall\, i \in [n]\big)\Big] \\ &= \mathbb{P}\big(g \geq -\sqrt{z}-t\big)^n = \big(1-\Phi(-\sqrt{z}-t)\big)^n. \end{aligned}$$

The pdf $f_X(z)$ of $X$ is the derivative of the cdf $F_X(z)$, so that, for $z \geq 0$, we obtain

$$f_X(z) = \frac{\mathrm{d}}{\mathrm{d}z}F_X(z) = \frac{n}{2\sqrt{z}}\cdot\big(1-\Phi(-\sqrt{z}-t)\big)^{n-1}\cdot\varphi(-\sqrt{z}-t).$$

Thus, the expected value $\mathbb{E}[\min\{0, \tilde{g}_n + t\}^2]$ can be estimated as

$$\begin{aligned}
\mathbb{E}\big[\min\{0, \tilde{g}_n + t\}^2\big] &= \int_0^\infty z \cdot \frac{n}{2\sqrt{z}} \cdot (1 - \Phi(-\sqrt{z} - t))^{n-1} \cdot \varphi(-\sqrt{z} - t)\, dz \\
&= n \cdot \int_{-\infty}^{-t} (x+t)^2 \cdot \varphi(x) \cdot \big(1 - \Phi(x)\big)^{n-1}\, dz \qquad (7.6) \\
&\leq n \cdot \int_{-\infty}^{-t} (x+t)^2 \cdot \varphi(x) \cdot \big(1 - \Phi(x)\big)\, dx \qquad (7.7) \\
&= n \cdot \int_{-\infty}^{-t} (x+t)^2 \cdot \varphi(x) \cdot \Phi(-x)\, dx. \qquad (7.8)
\end{aligned}$$

The first equality follows from the definition of the expected value; the second equality is due to a variable change $x = -\sqrt{z} - t$ and the inequality in (7.7) holds since $1 - \Phi(x) \leq 1$. In order to evaluate the integral in (7.8), we use the integrals in Owen [191]. This yields

$$\begin{aligned}
\int x^2 \cdot \varphi(x) \cdot \Phi(-x)\, dx &= -x\varphi(x)\Phi(-x) + \Phi(x) - \tfrac{1}{2}\Phi(x)^2 + \frac{1}{2\sqrt{2\pi}}\varphi(\sqrt{2}x), \\
\int x \cdot \varphi(x) \cdot \Phi(-x)\, dx &= -\frac{1}{2\sqrt{\pi}}\Phi(\sqrt{2}x) - \varphi(x)\Phi(-x), \\
\int \varphi(x) \cdot \Phi(-x)\, dx &= \Phi(x) - \tfrac{1}{2}\Phi(x)^2.
\end{aligned}$$

The desired expectation $\mathbb{E}[\min\{0, \tilde{g}_n + t\}^2]$ can now be computed as

$$\begin{aligned}
\mathbb{E}\big[\min\{0, \tilde{g}_n + t\}^2\big] \leq n\Big[&\big(1 + t^2\big)\big(\Phi(-t) - \tfrac{1}{2}\Phi(-t)^2\big) - t\varphi(-t)\Phi(t) \\
&- \frac{t}{\sqrt{\pi}}\Phi(-t\sqrt{2}) + \frac{1}{2\sqrt{2\pi}}\varphi(-t\sqrt{2})\Big],
\end{aligned}$$

so that $E_2^{(\mathrm{nng})}$ can be estimated as

$$\begin{aligned}
E_2^{(\mathrm{nng})} &= \mathbb{E}\Big[\min_{z_{s+1},\dots,z_n \leq t} \big(\sum_{i=s+1}^{n} (\tilde{g}_i + z_i)^2\big)^{1/2}\Big] \leq \Big((n-s) \cdot \mathbb{E}\big[\min\{0, \tilde{g}_n + t\}^2\big]\Big)^{1/2} \\
&\leq \Big[n(n-s) \cdot \Big(\big(1 + t^2\big)\big(\Phi(-t) - \tfrac{1}{2}\Phi(-t)^2\big) - t\varphi(-t)\Phi(t) - \frac{t}{\sqrt{\pi}}\Phi(-t\sqrt{2}) \\
&\qquad + \frac{1}{2\sqrt{2\pi}}\varphi(-t\sqrt{2})\Big)\Big]^{1/2}.
\end{aligned}$$

**Estimation of the Gaussian Width** Putting the estimations of $E_1^{(\mathrm{nng})}$ and $E_2^{(\mathrm{nng})}$ together yields the following estimate of the Gaussian width $\omega(T_s)$ which is valid for any (fixed) $t > 0$:

$$
\begin{aligned}
&\omega(T_s)\\
&\leq \mathbb{E}\Big[\min\Big\{\big(\sum_{i=1}^{s}(\tilde{g}_i + t)^2\big)^{1/2} + \big(\sum_{i=s+1}^{n}(\tilde{g}_i + z_i)^2\big)^{1/2} \;:\; t > 0,\; z_{s+1}, \dots, z_n \leq t\Big\}\Big]\\
&= E_1^{(\mathrm{nng})} + E_2^{(\mathrm{nng})}\\
&\leq t\sqrt{s} + \sqrt{\min_{\kappa>0}\Big\{(2+2\kappa)\Big[s\ln\big(\tfrac{en}{s}\big) + n\ln\big(\tfrac{1}{2}\big) + n\ln\big(1+\sqrt{1+\tfrac{1}{\kappa}}\big)\Big]\Big\} + \tfrac{1}{2}s}\\
&\quad + \Big(n(n-s)\cdot\big[\big(1+t^2\big)\big(\Phi(-t) - \tfrac{1}{2}\Phi(-t)^2\big) - t\varphi(-t)\Phi(t) - \frac{t}{\sqrt{\pi}}\Phi(-t\sqrt{2}) \qquad (7.9)\\
&\qquad\qquad + \frac{1}{2\sqrt{2\pi}}\varphi(-t\sqrt{2})\big]\Big)^{1/2}.
\end{aligned}
$$

Let $\omega(t)$ be the quantity in (7.9). Thus, the best bound on the Gaussian width is given by $\min_t \omega(t)$. Since we have derived a bound for the Gaussian width $\omega(T_s)$, we can now use Gordon's Escape Theorem 4.4 to prove Theorem 4.7.

*Proof of Theorem 4.7.* Let $t = \sqrt{2\ln(\frac{1}{\varepsilon})}$. Then, Gordon's Escape Theorem 4.4 yields

$$\mathbb{P}\Big(\inf_{x\in T}\|Ax\|_2 \leq E_m - \omega(T) - \sqrt{2\ln(\tfrac{1}{\varepsilon})}\Big) \leq \varepsilon,$$

where $T_s$ is defined in (4.8). Recall that $E_m$ is the expectation of the $\ell_2$-norm of a standard Gaussian random vector as defined in (4.1). Let $\omega = \min_t \omega(t)$ be the best estimation of $\omega(T_s)$ in (7.9). Since, by assumption,

$$\frac{m}{\sqrt{m+1}} \geq \omega + \sqrt{2\ln(\tfrac{1}{\varepsilon})},$$

and $E_m \geq m/\sqrt{m+1}$ we have $E_m - \omega(T) - \sqrt{2\ln(\frac{1}{\varepsilon})} \geq 0$. This implies

$$\mathbb{P}\big(\inf_{x\in T}\|Ax\|_2 \leq 0\big) \leq \mathbb{P}\big(\inf_{x\in T}\|Ax\|_2 \leq E_m - \omega(T_s \cap \mathbb{S}^{n-1}) - \sqrt{2\ln(\tfrac{1}{\varepsilon})}\big) \leq \varepsilon,$$

so that $\mathbb{P}\big(\inf_{x\in T}\|Ax\|_2 > 0\big) \geq 1-\varepsilon$, which shows that the nonnegative null space property ($\mathrm{NSP}_{\geq 0}$) of order $s$ is satisfied with probability at least $1-\varepsilon$. Thus, every $s$-sparse nonnegative $x \in \mathbb{R}^n_+$ is the unique optimal solution of the nonnegative $\ell_1$-minimization problem $\min\{\|z\|_1 \;:\; Az = Ax,\, z \geq 0\}$ with probability at least $1-\varepsilon$. □

# B Computational Results for MISDP Presolving

This section lists additional tables for the computational results on presolving for general MISDPs discussed in Section 6.6.3. In Table 6.3, we presented results over all 185 instances of the testset described in Section 6.6.1 for the settings listed in Section 6.6.2. Tables 7.1 to 7.5 in this chapter present these results for instance class separately. Shown are the number of instances that were solved to optimality within the time limit of one hour out of all 185 instances (# opt), and the shifted geometric means of the number of nodes (# nodes) as well as the CPU time in seconds (time), see (1.2) for the definition of the shifted geometric mean. The next columns list the shifted geometric mean of the CPU time in seconds used for presolving (time), the arithmetic mean of the number of domain reductions (# reds), i.e., changed bounds, and added constraints (# addcons) in presolving for SDP constraints. The section "SDP Constraints" in Table 6.3 shows the arithmetic means of the number of propagation calls (# prop), domain reductions (# reds), applied cuts (# cuts) and cutoffs (# cutoff) from SDP constraints. The last section "SDP Timings" shows the shifted geometric means of the the total time (total) and the propagation time (prop) spent for SDP constraints. For the shifted geometric means, we used a shift of $s = 100$ for nodes and $s = 1$ seconds for time, respectively.

**Table 7.1.** Comparison of presolving routines using the SDP- and LP-based approach for the 43 Cardinality Constrained Least Squares (CLS) instances.

| | | | | SDP presolving | | | SDP constraints | | | | SDP timings | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | time | #reds | #addcons | #prop | #reds | #cuts | #cutoff | total | prop |
| nopresol | 41 | 382.5 | 201.05 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.01 |
| DGZ | 41 | 362.4 | 206.48 | 0.00 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.01 |
| DZI | 41 | 382.3 | 200.63 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.01 |
| TM | 41 | 382.5 | 201.18 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.01 |
| TB-Pre | 41 | 382.4 | 200.02 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.01 |
| 2ML | 41 | 382.5 | 200.80 | 0.03 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.01 |
| 2MP | 41 | 382.2 | 200.59 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.01 |
| 2MV | 41 | 381.1 | 206.32 | 0.15 | 0.0 | 39 672.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.01 |
| PropUB-Pre | 41 | 382.7 | 201.29 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.01 |
| PropUB | 41 | 382.6 | 200.65 | 0.01 | 0.0 | 0.0 | 36 176.0 | 0.0 | 0.0 | 0.0 | 0.05 | 0.02 |
| PropTB | 41 | 334.2 | 99.87 | 0.00 | 0.0 | 0.0 | 3813.9 | 6.9 | 0.0 | 0.2 | 1.45 | 1.44 |
| MIX1 | 41 | 334.1 | 111.51 | 0.16 | 0.0 | 39 672.1 | 35 818.3 | 7.3 | 0.0 | 0.3 | 2.21 | 2.21 |
| MIX1-NoCA | 41 | 334.1 | 111.45 | 0.16 | 0.0 | 39 672.1 | 35 830.5 | 7.3 | 0.0 | 0.3 | 2.21 | 2.19 |
| MIX2 | 41 | 362.4 | 206.34 | 0.02 | 1.0 | 0.0 | 30 259.4 | 0.0 | 0.0 | 0.0 | 0.04 | 0.02 |
| allpresol | 41 | 380.6 | 205.72 | 0.19 | 1.0 | 39 672.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.01 |
| allprop | 41 | 334.0 | 110.15 | 0.01 | 0.0 | 0.0 | 35 772.5 | 6.9 | 0.0 | 0.7 | 3.12 | 3.11 |
| allprop-DGZ | 41 | 317.2 | 118.62 | 0.01 | 1.0 | 0.0 | 30 023.5 | 7.4 | 0.0 | 0.7 | 3.65 | 3.63 |
| allpresol-prop | 41 | 334.3 | 121.31 | 0.19 | 1.0 | 39 672.1 | 35 934.0 | 7.3 | 0.0 | 0.8 | 3.66 | 3.65 |
| LPA-nopresol | 43 | 201.1 | 7.53 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 | 2.0 | 3.65 | 0.00 |
| LPA-DGZ | 43 | 207.2 | 9.08 | 0.01 | 1.0 | 0.0 | 0.0 | 0.0 | 2992.2 | 4.8 | 3.90 | 0.00 |
| LPA-DZI | 43 | 201.1 | 7.70 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 | 2.0 | 3.70 | 0.00 |
| LPA-TM | 43 | 201.1 | 7.56 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 | 2.0 | 3.71 | 0.00 |
| LPA-TB-Pre | 43 | 201.1 | 7.76 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 | 2.0 | 3.72 | 0.00 |
| LPA-2ML | 43 | 201.1 | 7.69 | 0.03 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 | 2.0 | 3.73 | 0.00 |
| LPA-2MP | 43 | 201.1 | 7.68 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 | 2.0 | 3.74 | 0.00 |
| LPA-2MV | 43 | 214.5 | 10.53 | 0.15 | 0.0 | 39 672.1 | 0.0 | 0.0 | 3246.3 | 4.7 | 4.05 | 0.00 |
| LPA-PropUB-Pre | 43 | 201.1 | 7.69 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 3197.3 | 2.0 | 3.71 | 0.00 |
| LPA-PropUB | 43 | 201.1 | 7.68 | 0.01 | 0.0 | 0.0 | 4036.7 | 0.0 | 3197.3 | 2.0 | 3.71 | 0.00 |
| LPA-PropTB | 43 | 211.3 | 28.09 | 0.00 | 0.0 | 0.0 | 901.7 | 10.3 | 3100.8 | 2.1 | 18.05 | 9.89 |
| LPA-MIX1 | 43 | 197.8 | 21.75 | 0.16 | 0.0 | 39 672.1 | 5643.1 | 25.7 | 3188.0 | 5.3 | 10.32 | 4.95 |
| LPA-MIX2 | 43 | 207.2 | 9.08 | 0.02 | 1.0 | 0.0 | 6658.1 | 0.0 | 2992.2 | 4.8 | 3.94 | 0.01 |
| LPA-MIX2-NoCA | 43 | 198.6 | 8.89 | 0.02 | 1.0 | 0.0 | 6009.3 | 0.0 | 2815.1 | 3.7 | 3.93 | 0.00 |
| LPA-allpresol | 43 | 211.4 | 10.57 | 0.20 | 1.0 | 39 672.1 | 0.0 | 0.0 | 3249.3 | 4.6 | 4.06 | 0.00 |
| LPA-allprop | 43 | 197.8 | 48.07 | 0.01 | 0.0 | 0.0 | 3974.3 | 13.6 | 3059.2 | 15.9 | 31.38 | 10.19 |
| LPA-allpresol-prop | 41 | 183.6 | 100.89 | 0.19 | 1.0 | 39 672.1 | 5564.5 | 16.4 | 3107.2 | 10.9 | 64.23 | 29.34 |
| LPE-MIX2 | 32 | 19 781.8 | 106.76 | 0.02 | 1.0 | 0.0 | 1 447 705.2 | 0.0 | 103 867.4 | 19 388.7 | 31.46 | 0.51 |
| CONC: MIX1+LPA-MIX2 | 43 | 181.3 | 46.04 | 0.02 | 1.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 |

**Table 7.2.** Comparison of presolving routines using the SDP- and LP-based approach for the 32 Minimum $k$-Partitioning (MkP) instances.

| | | | | SDP presolving | | | SDP constraints | | | | SDP timings | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | time | #reds | #addcons | #prop | #reds | #cuts | #cutoff | total | prop |
| `nopresol` | 32 | 181.5 | 63.23 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.05 | 0.03 |
| `DGZ` | 32 | 181.5 | 63.23 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.03 |
| `DZI` | 32 | 181.5 | 63.23 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.03 |
| `TM` | 32 | 181.5 | 63.20 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.03 |
| `TB-Pre` | 32 | 181.5 | 63.18 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.02 |
| `2ML` | 32 | 181.5 | 63.22 | 0.02 | 0.0 | 110.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.02 |
| `2MP` | 32 | 181.5 | 63.19 | 0.02 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.02 |
| `2MV` | 32 | 181.5 | 63.42 | 0.03 | 0.0 | 2052.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.02 |
| `PropUB-Pre` | 32 | 181.5 | 63.33 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.03 |
| `PropUB` | 32 | 181.5 | 64.14 | 0.01 | 0.0 | 0.0 | 209 081.2 | 0.0 | 0.0 | 0.0 | 0.89 | 0.88 |
| `PropTB` | 32 | 181.5 | 63.18 | 0.00 | 0.0 | 0.0 | 499.4 | 0.0 | 0.0 | 0.0 | 0.04 | 0.03 |
| `MIX1` | 32 | 181.5 | 63.98 | 0.04 | 0.0 | 2052.2 | 209 081.2 | 0.0 | 0.0 | 0.0 | 0.91 | 0.90 |
| `MIX1-NoCA` | 32 | 182.0 | 64.68 | 0.04 | 0.0 | 2052.2 | 209 119.7 | 0.0 | 0.0 | 0.0 | 0.90 | 0.89 |
| `MIX2` | 32 | 181.5 | 63.86 | 0.01 | 0.0 | 0.0 | 209 081.2 | 0.0 | 0.0 | 0.0 | 0.89 | 0.88 |
| `allpresol` | 32 | 181.5 | 63.48 | 0.07 | 0.0 | 2162.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.03 |
| `allprop` | 32 | 181.5 | 63.94 | 0.01 | 0.0 | 0.0 | 209 081.2 | 0.0 | 0.0 | 0.0 | 0.88 | 0.87 |
| `allprop-DGZ` | 32 | 181.5 | 64.01 | 0.01 | 0.0 | 0.0 | 209 081.2 | 0.0 | 0.0 | 0.0 | 0.90 | 0.89 |
| `allpresol-prop` | 32 | 181.5 | 63.92 | 0.08 | 0.0 | 2162.9 | 209 081.2 | 0.0 | 0.0 | 0.0 | 0.88 | 0.87 |
| `LPA-nopresol` | 5 | 67.3 | 2737.16 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 24 934.4 | 0.0 | 1.42 | 0.00 |
| `LPA-DGZ` | 5 | 67.3 | 2740.72 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 24 940.7 | 0.0 | 1.42 | 0.01 |
| `LPA-DZI` | 4 | 67.5 | 2734.94 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 24 995.8 | 0.0 | 1.42 | 0.00 |
| `LPA-TM` | 5 | 67.3 | 2734.44 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 24 910.2 | 0.0 | 1.43 | 0.00 |
| `LPA-TB-Pre` | 4 | 67.5 | 2736.70 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 24 988.4 | 0.0 | 1.45 | 0.00 |
| `LPA-2ML` | 5 | 67.5 | 2735.28 | 0.02 | 0.0 | 110.6 | 0.0 | 0.0 | 24 979.4 | 0.0 | 1.46 | 0.00 |
| `LPA-2MP` | 5 | 67.5 | 2735.11 | 0.02 | 0.0 | 0.0 | 0.0 | 0.0 | 25 004.1 | 0.0 | 1.45 | 0.00 |
| `LPA-2MV` | 5 | 67.7 | 2734.81 | 0.03 | 0.0 | 2052.2 | 0.0 | 0.0 | 24 892.2 | 0.0 | 1.44 | 0.00 |
| `LPA-PropUB-Pre` | 4 | 67.3 | 2735.38 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 24 936.4 | 0.0 | 1.42 | 0.01 |
| `LPA-PropUB` | 4 | 67.2 | 2736.93 | 0.01 | 0.0 | 0.0 | 26 071.9 | 0.0 | 24 942.1 | 0.0 | 1.77 | 0.31 |
| `LPA-PropTB` | 5 | 67.6 | 2735.21 | 0.00 | 0.0 | 0.0 | 194.1 | 0.0 | 25 012.2 | 0.0 | 1.41 | 0.00 |
| `LPA-MIX1` | 4 | 67.4 | 2738.49 | 0.04 | 0.0 | 2052.2 | 26 065.8 | 0.0 | 24 915.2 | 0.0 | 1.77 | 0.30 |
| `LPA-MIX2` | 4 | 67.3 | 2737.89 | 0.01 | 0.0 | 0.0 | 26 046.8 | 0.0 | 24 934.2 | 0.0 | 1.79 | 0.29 |
| `LPA-MIX2-NoCA` | 5 | 57.5 | 2408.40 | 0.01 | 0.0 | 0.0 | 24 964.9 | 0.0 | 22 153.9 | 0.0 | 1.61 | 0.28 |
| `LPA-allpresol` | 4 | 67.5 | 2736.88 | 0.07 | 0.0 | 2162.9 | 0.0 | 0.0 | 24 965.7 | 0.0 | 1.44 | 0.00 |
| `LPA-allprop` | 5 | 67.4 | 2735.64 | 0.01 | 0.0 | 0.0 | 26 083.1 | 0.0 | 24 958.2 | 0.0 | 1.80 | 0.30 |
| `LPA-allpresol-prop` | 5 | 67.3 | 2739.24 | 0.08 | 0.0 | 2162.9 | 26 056.8 | 0.0 | 24 913.6 | 0.0 | 1.77 | 0.30 |
| `LPE-MIX2` | 1 | 731 548.2 | 3074.35 | 0.01 | 0.0 | 0.0 | 2 856 042.9 | 0.0 | 62 998.1 | 0.0 | 30.51 | 27.70 |
| `CONC: MIX1+LPA-MIX2` | 29 | 168.8 | 93.52 | 0.01 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 |

**Table 7.3.** Comparison of presolving routines using the SDP- and LP-based approach for the 46 Restricted Isometry Property (RIP) instances.

| | | | | SDP presolving | | | SDP constraints | | | | SDP timings | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | time | #reds | #addcons | #prop | #reds | #cuts | #cutoff | total | prop |
| nopresol | 36 | 4376.2 | 259.70 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.08 | 0.04 |
| DGZ | 36 | 4443.0 | 296.45 | 0.00 | 43.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.08 | 0.05 |
| DZI | 36 | 4376.3 | 259.95 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.08 | 0.05 |
| TM | 36 | 4369.4 | 260.17 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.09 | 0.05 |
| TB-Pre | 36 | 4376.1 | 259.73 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.08 | 0.05 |
| 2ML | 36 | 4173.6 | 281.78 | 0.02 | 0.0 | 994.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.07 | 0.04 |
| 2MP | 36 | 4371.5 | 259.52 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.07 | 0.05 |
| 2MV | 36 | 4376.2 | 261.18 | 0.04 | 0.0 | 1988.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.08 | 0.05 |
| PropUB-Pre | 36 | 2755.9 | 198.09 | 0.01 | 1988.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.07 | 0.04 |
| PropUB | 36 | 2756.7 | 199.05 | 0.01 | 1988.3 | 0.0 | 184 929.1 | 0.2 | 0.0 | 0.0 | 1.58 | 1.56 |
| PropTB | 36 | 4377.1 | 259.67 | 0.00 | 0.0 | 0.0 | 25 049.6 | 0.0 | 0.0 | 0.0 | 0.07 | 0.05 |
| MIX1 | 36 | 2759.5 | 194.11 | 0.04 | 1988.3 | 1988.3 | 186 004.4 | 0.2 | 0.0 | 0.0 | 1.62 | 1.60 |
| MIX1-NoCA | 36 | 2761.9 | 193.71 | 0.04 | 1988.3 | 1988.3 | 186 254.2 | 0.2 | 0.0 | 0.0 | 1.58 | 1.56 |
| MIX2 | 36 | 2757.6 | 225.43 | 0.01 | 2031.5 | 0.0 | 183 489.4 | 5492.2 | 0.0 | 0.0 | 1.58 | 1.56 |
| allpresol | 36 | 2768.3 | 227.89 | 0.06 | 2031.5 | 2982.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.06 | 0.03 |
| allprop | 36 | 2755.0 | 199.16 | 0.01 | 1988.3 | 0.0 | 184 592.3 | 0.2 | 0.0 | 0.0 | 1.59 | 1.57 |
| allprop-DGZ | 36 | 2755.7 | 225.17 | 0.01 | 2031.5 | 0.0 | 183 360.7 | 5492.2 | 0.0 | 0.0 | 1.59 | 1.57 |
| allpresol-prop | 36 | 2757.7 | 225.37 | 0.06 | 2031.5 | 2982.4 | 183 484.5 | 5017.5 | 0.0 | 0.2 | 1.59 | 1.57 |
| LPA-nopresol | 0 | 35.7 | 3600.32 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 12 779.8 | 0.0 | 0.61 | 0.00 |
| LPA-DGZ | 0 | 36.7 | 3600.51 | 0.00 | 43.3 | 0.0 | 0.0 | 0.0 | 13 240.9 | 0.0 | 0.64 | 0.00 |
| LPA-DZI | 0 | 35.9 | 3600.40 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 12 829.6 | 0.0 | 0.60 | 0.00 |
| LPA-TM | 0 | 36.3 | 3600.42 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 12 991.6 | 0.0 | 0.64 | 0.00 |
| LPA-TB-Pre | 0 | 35.9 | 3600.58 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 12 895.0 | 0.0 | 0.61 | 0.00 |
| LPA-2ML | 0 | 37.4 | 3600.38 | 0.02 | 0.0 | 994.1 | 0.0 | 0.0 | 13 218.0 | 0.0 | 0.65 | 0.00 |
| LPA-2MP | 0 | 36.2 | 3600.51 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 12 924.5 | 0.0 | 0.63 | 0.00 |
| LPA-2MV | 0 | 36.3 | 3600.70 | 0.03 | 0.0 | 1988.3 | 0.0 | 0.0 | 12 928.0 | 0.0 | 0.62 | 0.00 |
| LPA-PropUB-Pre | 0 | 30.1 | 3600.77 | 0.01 | 1988.3 | 0.0 | 0.0 | 0.0 | 13 507.7 | 0.0 | 0.62 | 0.00 |
| LPA-PropUB | 0 | 29.9 | 3600.90 | 0.01 | 1988.3 | 0.0 | 10 355.9 | 0.0 | 13 433.2 | 0.0 | 0.74 | 0.10 |
| LPA-PropTB | 0 | 35.8 | 3600.48 | 0.00 | 0.0 | 0.0 | 122.1 | 0.0 | 12 870.5 | 0.0 | 0.62 | 0.00 |
| LPA-MIX1 | 0 | 29.5 | 3600.40 | 0.04 | 1988.3 | 1988.3 | 10 927.8 | 0.0 | 13 673.1 | 0.0 | 0.75 | 0.10 |
| LPA-MIX2 | 0 | 29.9 | 3600.64 | 0.01 | 2031.5 | 0.0 | 10 646.2 | 0.0 | 13 778.3 | 0.0 | 0.77 | 0.10 |
| LPA-MIX2-NoCA | 0 | 30.0 | 3600.70 | 0.01 | 2031.5 | 0.0 | 10 652.8 | 0.0 | 13 777.7 | 0.0 | 0.76 | 0.10 |
| LPA-allpresol | 0 | 32.2 | 3600.56 | 0.06 | 2031.5 | 2982.4 | 0.0 | 0.0 | 14 095.3 | 0.0 | 0.69 | 0.00 |
| LPA-allprop | 0 | 29.8 | 3600.60 | 0.01 | 1988.3 | 0.0 | 10 393.3 | 0.0 | 13 402.5 | 0.0 | 0.74 | 0.09 |
| LPA-allpresol-prop | 0 | 32.3 | 3600.76 | 0.06 | 2031.5 | 2982.4 | 10 920.9 | 0.0 | 14 131.2 | 0.0 | 0.81 | 0.11 |
| LPE-MIX2 | 33 | 41 373.9 | 366.98 | 0.01 | 2031.5 | 0.0 | 532 210.1 | 359 368.7 | 131 484.5 | 64.8 | 7.02 | 4.27 |
| CONC: MIX1+LPA-MIX2 | 30 | 1371.2 | 513.69 | 0.01 | 2031.5 | 0.0 | 79.6 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 |

**Table 7.4.** Comparison of presolving routines using the SDP- and LP-based approach for the 26 random MISDP (RND) instances.

| | | | | | SDP presolving | | | SDP constraints | | | SDP timings | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | time | #reds | #addcons | #prop | #reds | #cuts | #cutoff | total | prop |
| `nopresol` | 25 | 98.4 | 268.58 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `DGZ` | 25 | 98.3 | 268.85 | 0.00 | 0.0 | 96.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.00 |
| `DZI` | 25 | 98.3 | 269.73 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `TM` | 25 | 98.3 | 268.78 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `TB-Pre` | 25 | 98.3 | 268.44 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.00 |
| `2ML` | 25 | 98.3 | 270.37 | 0.03 | 0.0 | 4797.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.04 | 0.00 |
| `2MP` | 25 | 98.3 | 268.78 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `2MV` | 25 | 98.3 | 269.02 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `PropUB-Pre` | 25 | 98.3 | 269.05 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `PropUB` | 25 | 98.3 | 268.69 | 0.01 | 0.0 | 0.0 | 2903.5 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `PropTB` | 25 | 98.3 | 269.00 | 0.00 | 0.0 | 0.0 | 120.6 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `MIX1` | 25 | 98.3 | 268.99 | 0.02 | 0.0 | 0.0 | 2903.5 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `MIX1-NoCA` | 25 | 98.3 | 269.15 | 0.02 | 0.0 | 0.0 | 2903.5 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `MIX2` | 25 | 98.4 | 269.03 | 0.01 | 0.0 | 96.9 | 2903.6 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `allpresol` | 25 | 98.3 | 270.26 | 0.05 | 0.0 | 4894.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `allprop` | 25 | 98.4 | 268.81 | 0.01 | 0.0 | 0.0 | 2903.6 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `allprop-DGZ` | 25 | 98.4 | 268.64 | 0.01 | 0.0 | 96.9 | 2903.6 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `allpresol-prop` | 25 | 98.3 | 270.46 | 0.05 | 0.0 | 4894.6 | 2903.5 | 0.0 | 0.0 | 0.0 | 0.03 | 0.00 |
| `LPA-nopresol` | 26 | 99.8 | 413.63 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 | 0.0 | 104.20 | 0.00 |
| `LPA-DGZ` | 25 | 97.6 | 423.27 | 0.00 | 0.0 | 96.9 | 0.0 | 0.0 | 43 599.4 | 0.0 | 105.26 | 0.00 |
| `LPA-DZI` | 26 | 99.8 | 409.12 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 | 0.0 | 102.85 | 0.00 |
| `LPA-TM` | 26 | 99.8 | 412.75 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 | 0.0 | 103.56 | 0.00 |
| `LPA-TB-Pre` | 26 | 99.8 | 412.29 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 | 0.0 | 103.50 | 0.00 |
| `LPA-2ML` | 25 | 96.0 | 417.89 | 0.03 | 0.0 | 4797.7 | 0.0 | 0.0 | 42 159.3 | 0.0 | 102.70 | 0.00 |
| `LPA-2MP` | 26 | 99.8 | 411.54 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 | 0.0 | 103.81 | 0.00 |
| `LPA-2MV` | 26 | 99.8 | 412.07 | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 | 0.0 | 103.29 | 0.00 |
| `LPA-PropUB-Pre` | 26 | 99.8 | 413.21 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 41 561.1 | 0.0 | 103.36 | 0.00 |
| `LPA-PropUB` | 26 | 99.8 | 413.63 | 0.00 | 0.0 | 0.0 | 2193.6 | 0.0 | 41 561.1 | 0.0 | 104.06 | 0.00 |
| `LPA-PropTB` | 26 | 99.8 | 412.04 | 0.00 | 0.0 | 0.0 | 161.3 | 0.0 | 41 561.1 | 0.0 | 103.65 | 0.00 |
| `LPA-MIX1` | 26 | 99.8 | 412.45 | 0.01 | 0.0 | 0.0 | 2193.6 | 0.0 | 41 561.1 | 0.0 | 103.39 | 0.00 |
| `LPA-MIX2` | 25 | 97.6 | 422.87 | 0.01 | 0.0 | 96.9 | 2064.0 | 0.0 | 43 527.3 | 0.0 | 105.44 | 0.00 |
| `LPA-MIX2-NoCA` | 25 | 98.6 | 418.66 | 0.01 | 0.0 | 96.9 | 1828.1 | 0.0 | 44 157.1 | 0.0 | 106.42 | 0.00 |
| `LPA-allpresol` | 25 | 95.4 | 419.32 | 0.05 | 0.0 | 4894.6 | 0.0 | 0.0 | 40 530.2 | 0.0 | 101.47 | 0.00 |
| `LPA-allprop` | 26 | 99.8 | 412.99 | 0.00 | 0.0 | 0.0 | 2193.6 | 0.0 | 41 561.1 | 0.0 | 103.68 | 0.00 |
| `LPA-allpresol-prop` | 25 | 95.4 | 418.40 | 0.05 | 0.0 | 4894.6 | 2191.8 | 0.0 | 40 589.2 | 0.0 | 101.18 | 0.00 |
| `LPE-MIX2` | 7 | 37 491.6 | 2232.65 | 0.01 | 0.0 | 96.9 | 325 125.5 | 0.0 | 78 329.4 | 0.0 | 133.61 | 0.09 |
| `CONC: MIX1+LPA-MIX2` | 25 | 67.6 | 179.92 | 0.01 | 0.0 | 96.9 | 75.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 |

**Table 7.5.** Comparison of presolving routines using the SDP- and LP-based approach for the 38 Truss Topology Design (TTD) instances.

| | | | | SDP presolving | | | SDP constraints | | | | SDP timings | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| setting | #opt | #nodes | time | time | #reds | #addcons | #prop | #reds | #cuts | #cutoff | total | prop |
| nopresol | 34 | 23 840.6 | 187.36 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.46 | 0.36 |
| DGZ | 34 | 23 839.7 | 187.16 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.48 | 0.34 |
| DZI | 34 | 17 547.3 | 146.73 | 0.00 | 0.0 | 3.6 | 0.0 | 0.0 | 0.0 | 0.0 | 1.23 | 0.26 |
| TM | 34 | 23 840.5 | 188.81 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.50 | 0.36 |
| TB-Pre | 33 | 23 700.8 | 191.85 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.52 | 0.38 |
| 2ML | 34 | 23 842.6 | 188.11 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.49 | 0.34 |
| 2MP | 34 | 23 839.8 | 188.01 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.45 | 0.34 |
| 2MV | 33 | 21 561.9 | 167.33 | 0.00 | 0.0 | 62.1 | 0.0 | 0.0 | 0.0 | 0.0 | 1.43 | 0.32 |
| PropUB-Pre | 34 | 23 841.9 | 188.00 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.47 | 0.34 |
| PropUB | 34 | 23 841.3 | 187.69 | 0.00 | 0.0 | 0.0 | 2 718 636.0 | 0.0 | 0.0 | 0.0 | 1.58 | 0.50 |
| PropTB | 33 | 18 695.5 | 182.75 | 0.00 | 0.0 | 0.0 | 83 737.6 | 16 440.6 | 0.0 | 1609.0 | 33.24 | 32.58 |
| MIX1 | 33 | 14 397.8 | 147.75 | 0.00 | 0.0 | 65.7 | 1 713 894.0 | 10 756.0 | 0.0 | 721.2 | 27.57 | 27.05 |
| MIX1-NoCA | 33 | 14 274.5 | 145.71 | 0.00 | 0.0 | 65.7 | 1 718 698.6 | 10 820.2 | 0.0 | 726.7 | 26.68 | 26.15 |
| MIX2 | 34 | 17 547.1 | 146.42 | 0.00 | 0.0 | 3.6 | 2 129 008.7 | 0.0 | 0.0 | 0.0 | 1.31 | 0.40 |
| allpresol | 34 | 18 398.9 | 151.30 | 0.00 | 0.0 | 65.7 | 0.0 | 0.0 | 0.0 | 0.0 | 1.27 | 0.30 |
| allprop | 32 | 12 454.4 | 252.35 | 0.00 | 0.0 | 0.0 | 578 865.9 | 20 123.6 | 0.0 | 14 549.5 | 151.89 | 151.85 |
| allprop-DGZ | 32 | 12 428.8 | 253.52 | 0.00 | 0.0 | 0.0 | 568 226.6 | 19 768.7 | 0.0 | 14 453.9 | 152.92 | 152.88 |
| allpresol-prop | 34 | 9293.6 | 192.47 | 0.00 | 0.0 | 65.7 | 440 865.0 | 21 242.3 | 0.0 | 9080.5 | 115.26 | 115.22 |
| LPA-nopresol | 30 | 17 646.4 | 210.55 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 155 497.6 | 527.1 | 5.20 | 0.07 |
| LPA-DGZ | 30 | 17 635.4 | 210.67 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 155 245.6 | 527.1 | 5.23 | 0.07 |
| LPA-DZI | 26 | 13 931.7 | 169.27 | 0.00 | 0.0 | 3.6 | 0.0 | 0.0 | 154 463.6 | 222.1 | 4.73 | 0.05 |
| LPA-TM | 30 | 17 628.5 | 212.69 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 155 079.4 | 527.1 | 5.18 | 0.07 |
| LPA-TB-Pre | 28 | 18 168.5 | 218.12 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 161 128.7 | 551.1 | 5.44 | 0.07 |
| LPA-2ML | 30 | 17 620.8 | 210.28 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 155 065.0 | 527.1 | 5.20 | 0.06 |
| LPA-2MP | 29 | 17 637.8 | 213.57 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 155 304.2 | 527.1 | 5.21 | 0.06 |
| LPA-2MV | 27 | 13 937.4 | 169.64 | 0.00 | 0.0 | 62.1 | 0.0 | 0.0 | 148 992.6 | 215.3 | 4.73 | 0.07 |
| LPA-PropUB-Pre | 30 | 17 617.8 | 210.71 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 154 897.7 | 527.1 | 5.18 | 0.07 |
| LPA-PropUB | 30 | 17 650.9 | 212.82 | 0.00 | 0.0 | 0.0 | 343 292.3 | 0.0 | 155 555.5 | 527.7 | 5.20 | 0.11 |
| LPA-PropTB | 29 | 14 602.5 | 231.70 | 0.00 | 0.0 | 0.0 | 72 529.8 | 22 910.9 | 170 561.6 | 2026.9 | 35.27 | 30.97 |
| LPA-MIX1 | 28 | 10 918.4 | 165.25 | 0.00 | 0.0 | 65.7 | 285 380.5 | 16 965.0 | 138 497.7 | 736.9 | 27.01 | 23.68 |
| LPA-MIX2 | 26 | 13 983.5 | 168.97 | 0.00 | 0.0 | 3.6 | 332 744.0 | 0.0 | 155 157.1 | 224.2 | 4.81 | 0.11 |
| LPA-MIX2-NoCA | 29 | 13 935.7 | 154.24 | 0.00 | 0.0 | 3.6 | 270 283.5 | 0.0 | 133 823.3 | 252.4 | 4.36 | 0.10 |
| LPA-allpresol | 27 | 14 548.4 | 163.72 | 0.00 | 0.0 | 65.7 | 0.0 | 0.0 | 152 821.3 | 230.0 | 4.75 | 0.06 |
| LPA-allprop | 31 | 13 857.6 | 311.69 | 0.00 | 0.0 | 0.0 | 277 894.1 | 15 537.3 | 124 126.5 | 4400.9 | 139.89 | 131.06 |
| LPA-allpresol-prop | 28 | 10 525.4 | 251.53 | 0.00 | 0.0 | 65.7 | 237 691.5 | 20 502.8 | 137 776.4 | 4300.1 | 114.22 | 106.39 |
| LPE-MIX2 | 11 | 421 128.0 | 938.85 | 0.00 | 0.0 | 3.6 | 1 920 622.5 | 0.0 | 348 792.6 | 15 301.8 | 13.80 | 0.70 |
| CONC: MIX1+LPA-MIX2 | 33 | 9125.8 | 93.54 | 0.00 | 0.0 | 3.6 | 91.5 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 |

# Bibliography

[1] T. Achterberg. *Constraint Integer Programming*. PhD thesis, TU Berlin, 2007. `http://opus.kobv.de/tuberlin/volltexte/2007/1611/`. [→11, 158]

[2] T. Achterberg. Conflict analysis in mixed integer programming. *Discrete Optimization*, 4(1):4–20, 2007. [→159]

[3] T. Achterberg and R. Wunderling. Mixed integer programming: Analyzing 12 years of progress. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization*, pages 449–481. Springer Berlin Heidelberg, 2013. [→152]

[4] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, and D. Weninger. Presolve reductions in mixed integer programming. *INFORMS Journal on Computing*, 32(2):473–506, 2020. [→158]

[5] B. Adcock, A. C. Hansen, C. Poon, and B. Roman. Breaking the coherence barrier: A new theory for compressed sensing. In *Forum of Mathematics, Sigma*, volume 5. Cambridge University Press, 2017. [→94]

[6] F. Affentranger and R. Schneider. Random projections of regular simplices. *Discrete & Computational Geometry*, 7(3):219–226, 1992. [→100]

[7] A. A. Ahmadi and G. Hall. DC decomposition of nonconvex polynomials with algebraic techniques. *Mathematical Programming*, 169(1):69–94, 2018. [→26]

[8] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1):237–260, 1998. [→4]

[9] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp. Living on the edge: Phase transitions in convex programs with random data. *Information and Inference: A Journal of the IMA*, 3(3):224–294, 2014. [→45, 46, 99, 100, 105, 121]

[10] K. Ardah, M. Haardt, T. Liu, F. Matter, M. Pesavento, and M. E. Pfetsch. Recovery under side constraints. Preprint, arXiv:2106.09375, 2021. [→3]

[11] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor. Spectrum sensing for cognitive radio : State-of-the-art and recent advances. *IEEE Signal Processing Magazine*, 29(3):101–116, 2012. [→66]

[12] A. Aïssa-El-Bey, D. Pastor, S. M. A. Sbaï, and Y. Fadlallah. Sparsity-based recovery of finite alphabet solutions to underdetermined linear systems. *IEEE Transactions on Information Theory*, 61(4):2008–2018, 2015. [→67]

[13] A. S. Bandeira, M. Fickus, D. G. Mixon, and P. Wong. The road to deterministic matrices with the restricted isometry property. *Journal of Fourier Analysis and Applications*, 19(6):1123–1149, 2013. [→97]

[14] R. G. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24 (4):118–121, 2007. [→3]

[15] R. G. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008. [→99, 176]

[16] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010. [→95]

[17] A. Bastounis and A. C. Hansen. On the absence of uniform recovery in many real-world applications of compressed sensing and the restricted isometry property and nullspace property in levels. *SIAM Journal on Imaging Sciences*, 10 (1):335–371, 2017. [→94, 95]

[18] M. Beauchamp. On numerical computation for the distribution of the convolution of N independent rectified Gaussian variables. *Journal de la Société Française de Statistique*, 159(1):88–111, 2018. [→207]

[19] P. Belotti, S. Cafieri, J. Lee, and L. Liberti. Feasibility-based bounds tightening via fixed points. In W. Wu and O. Daescu, editors, *Combinatorial Optimization and Applications – 4th International Conference, COCOA 2010*, volume 6508 of *Lecture Notes in Computer Science*, pages 65–76. Springer, 2010. [→167, 168]

[20] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013. [→158]

[21] A. Ben-Tal and A. Nemirovski. Robust truss topology design via semidefinite programming. *SIAM Journal on Optimization*, 7(4):991–1016, 1997. [→177]

[22] A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26:193–205, 2001. [→157]

[23] D. Bertsimas and B. Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, 48(1): 300–323, 2020. [→175]

[24] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. Solving large-scale sparse PCA to certifiable (near) optimality. *Journal of Machine Learning Research*, 23(13):1–35, 2022. [→143, 144, 195]

[25] K. Bestuzheva, M. Besançon, W.-K. Chen, A. Chmiela, T. Donkiewicz, J. van Doornmalen, L. Eifler, O. Gaul, G. Gamrath, A. Gleixner, L. Gottwald, C. Graczyk, K. Halbig, A. Hoen, C. Hojny, R. van der Hulst, T. Koch, M. Lübbecke, S. J. Maher, F. Matter, E. Mühmer, B. Müller, M. E. Pfetsch, D. Rehfeldt, S. Schlein, F. Schlösser, F. Serrano, Y. Shinano, B. Sofranac, M. Turner, S. Vigerske, F. Wegscheider, P. Wellner, D. Weninger, and J. Witzig. The SCIP Optimization Suite 8.0. ZIB-Report 21-41, Zuse Institute Berlin, 2021. `http://nbn-resolving.de/urn:nbn:de:0297-zib-85309`. [→7, 155]

[26] R. Bixby and E. Rothberg. Progress in computational mixed integer programming—A look back from the other side of the tipping point. *Annals of Operations Research*, 149:37–41, 2007. [→152]

[27] G. Blekherman, S. S. Dey, M. Molinaro, and S. Sun. Sparse PSD approximation of the PSD cone. *Mathematical Programming*, pages 1–24, 2020. [→148]

[28] T. Blumensath and M. E. Davies. Sampling theorems for signals from the union of finite-dimensional linear subspaces. *IEEE Transactions on Information Theory*, 55(4):1872–1882, 2009. [→52, 95]

[29] L. Bordeaux, G. Katsirelos, N. Narodytska, and M. Y. Vardi. The complexity of integer bound propagation. *Journal of Artificial Intelligence Research*, 40: 657–676, 2011. [→168]

[30] K. Böröczky and M. Henk. Random projections of regular polytopes. *Archiv der Mathematik*, 73(6):465–473, 1999. [→100]

[31] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004. [→10]

[32] G. Braun, S. Fiorini, S. Pokutta, and D. Steurer. Approximation limits of linear programs (beyond hierarchies). *Mathematics of Operations Research*, 40(3):756–772, 2015. [→157]

[33] G. Braun, S. Pokutta, and D. Zink. Affine reductions for LPs and SDPs. *Mathematical Programming*, 173:281–312, 2019. [→157]

[34] A. L. Brearley, G. Mitra, and H. P. Williams. Analysis of mathematical programming problems prior to applying the simplex algorithm. *Mathematical Programming*, 8(1):54–83, 1975. [→158]

[35] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009. [→2]

[36] J.-F. Cai and W. Xu. Guarantees of total variation minimization for signal recovery. *Information and Inference: A Journal of the IMA*, 4(4):328–353, 2015. [→15]

[37] T. T. Cai and A. Zhang. Sparse representation of a polytope and recovery of sparse signals and low-rank matrices. *IEEE Transactions on Information Theory*, 60(1):122–132, 2014. [→142]

[38] E. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. [→2, 5, 99, 142]

[39] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. [→2, 6, 29, 142]

[40] E. J. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematicians (ICM)*, volume 3, pages 1433–1452. European Mathematical Society, Madrid, Spain, 2006. [→3]

[41] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008. [→142]

[42] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise $C^2$ singularities. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(2):219–266, 2004. [→15]

[43] E. J. Candès and Y. Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory*, 57(4):2342–2359, 2011. [→29]

[44] E. J. Candès and B. Recht. Simple bounds for recovering low-complexity models. *Mathematical Programming*, 141(1):577–589, 2013. [→6, 19, 99, 121, 200]

[45] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006. [→5, 99, 142]

[46] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008. [→3]

[47] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, 31(1):59–73, 2011. [→15]

[48] T. F. Chan and J. J. Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*, volume 94. SIAM, 2005. [→15]

[49] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012. [→6, 45, 46, 47, 99, 100, 103, 105, 121, 200]

[50] J. Chen and X. Huo. Theoretical results on sparse representations of multiple-measurement vectors. *IEEE Transactions on Signal Processing*, 54(12):4634–4643, 2006. [→52]

[51] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001. [→4]

[52] J. W. Chinneck. *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*, volume 118 of *International Series in Operations Research and Management Sciences.* Springer, 2008. [→54]

[53] M. Cho, K. Vijay Mishra, and W. Xu. Computable performance guarantees for compressed sensing matrices. *EURASIP Journal on Advances in Signal Processing*, 2018(1):1–18, 2018. [→125]

[54] C. Coey, M. Lubin, and J. P. Vielma. Outer approximation with conic certificates for mixed-integer convex problems. *Mathematical Programming Computation*, 12:249–293, 2020. [→155]

[55] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best $k$-term approximation. *Journal of the American Mathematical Society*, 22(1):211–231, 2009. [→5, 13, 62]

[56] S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, 53(7):2477–2488, 2005. [→52]

[57] H. Crowder, E. L. Johnson, and M. Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31:803–834, 1983. [→158]

[58] R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965. [→11, 154]

[59] A. d'Aspremont and L. El Ghaoui. Testing the nullspace property using semidefinite programming. *Mathematical Programming*, 127(1):123–144, 2011. [→123, 125, 128, 129]

[60] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007. [→195]

[61] M. A. Davenport and J. Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016. [→99]

[62] R. A. DeVore. Deterministic constructions of compressed sensing matrices. *Journal of Complexity*, 23(4):918–925, 2007. [→97]

[63] S. S. Dey and M. Molinaro. Theoretical challenges towards cutting-plane selection. *Mathematical Programming*, 170(1):237–266, 2018. [→148]

[64] S. S. Dey, A. M. Kazachkov, A. Lodi, and G. Munoz. Cutting plane generation through sparse principal component analysis. Preprint, Optimization Online, 2021. `http://www.optimization-online.org/DB_HTML/2021/02/8259.html`. [→148]

[65] S. S. Dey, R. Mazumder, and G. Wang. Using $\ell_1$-relaxation and integer programming to obtain dual bounds for sparse PCA. *Operations Research*, 70(3): 1914–1932, 2021. [→195]

[66] S. Dirksen, G. Lecué, and H. Rauhut. On the gap between restricted isometry properties and sparse recovery conditions. *IEEE Transactions on Information Theory*, 64(8):5478–5487, 2018. [→99, 201]

[67] D. L. Donoho. Neighborly polytopes and sparse solutions of underdetermined linear equations. Technical Report 2005-4, Dept. of Statistics, Stanford Univ., 2005. [→64, 100]

[68] D. L. Donoho. High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension. *Discrete & Computational Geometry*, 35 (4):617–652, 2006. [→100]

[69] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. [→2]

[70] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell_1$ minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003. [→5]

[71] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, 2001. [→5, 62]

[72] D. L. Donoho and B. F. Logan. Signal recovery and the large sieve. *SIAM Journal on Applied Mathematics*, 52(2):577–591, 1992. [→4]

[73] D. L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences*, 102(27):9446–9451, 2005. [→64, 100]

[74] D. L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. Technical Report 2005-6, Dept. of Statistics, Stanford Univ., 2005. [→64, 100]

[75] D. L. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences*, 102(27): 9452–9457, 2005. [→100]

[76] D. L. Donoho and J. Tanner. Thresholds for the recovery of sparse solutions via l1 minimization. In *2006 40th Annual Conference on Information Sciences and Systems*, pages 202–206, 2006. [→100]

[77] D. L. Donoho and J. Tanner. Counting faces of randomly projected polytopes when the projection radically lowers dimension. *Journal of the American Mathematical Society*, 22(1):1–53, 2009. [→100]

[78] D. L. Donoho and J. Tanner. Precise undersampling theorems. *Proceedings of the IEEE*, 98(6):913–924, 2010. [→100]

[79] D. L. Donoho and J. Tanner. Counting the faces of randomly-projected hypercubes and orthants, with applications. *Discrete & Computational Geometry*, 43(3):522–541, 2010. [→100]

[80] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36: 307–339, 1986. [→155]

[81] A. Eisenblätter. *Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds.* PhD thesis, TU Berlin, 2001. [→175]

[82] A. Eisenblätter. The semidefinite relaxation of the $k$-partition polytope is strong. In W. J. Cook and A. S. Schulz, editors, *Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization*, volume 2337 of *Lecture Notes in Computer Science*, pages 273–290. Springer, Berlin Heidelberg, 2002. [→175]

[83] J. Eisert, A. Flinth, B. Groß, I. Roth, and G. Wunder. Hierarchical compressed sensing. Preprint, arXiv:2104.02721, 2021. [→54, 94]

[84] M. Elad. *Sparse and redundant representations: From theory to applications in signal and image processing.* Springer, 2010. [→2]

[85] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15 (12):3736–3745, 2006. [→2]

[86] M. Elad and A. Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002. [→5]

[87] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. *Inverse Problems*, 23(3):947–968, 2007. [→15]

[88] Y. C. Eldar and G. Kutyniok. *Compressed sensing: Theory and applications.* Cambridge University Press, 2012. [→3]

[89] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Transactions on Information Theory*, 55(11):5302–5316, 2009. [→52, 95, 202]

[90] Y. C. Eldar, P. Kuppinger, and H. Bölcskei. Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing*, 58 (6):3042–3054, 2010. [→52]

[91] E. Elhamifar and R. Vidal. Block-sparse recovery via convex optimization. *IEEE Transactions on Signal Processing*, 60(8):4094–4107, 2012. [→52]

[92] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. [→53]

[93] F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems – Volume II.* Springer, 2003. [→171]

[94] M. Fazel. *Matrix Rank Minimization with Applications.* PhD thesis, Stanford University, 2002. [→13]

[95] T. Fischer and M. E. Pfetsch. Monoidal cut strengthening and generalized mixed-integer rounding for disjunctive programs. *Operations Research Letters*, 45(6):556–560, 2017. [→72]

[96] T. Fischer and M. E. Pfetsch. Branch-and-cut for linear programs with overlapping SOS1 constraints. *Mathematical Programming Computation*, 10(1): 33–68, 2018. [→72, 129]

[97] T. Fischer, G. Hegde, F. Matter, M. Pesavento, M. E. Pfetsch, and A. M. Tillmann. Joint antenna selection and phase-only beamforming using mixed-integer nonlinear programming. In *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, pages 1–7, 2018. [→9, 52, 78, 79, 80, 90, xi]

[98] A. Flinth and S. Keiper. Recovery of binary sparse signals with biased measurement matrices. *IEEE Transactions on Information Theory*, 65(12):8084–8094, 2019. [→67]

[99] M. Fornasier and H. Rauhut. Compressive sensing. In O. Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, pages 187–228. Springer New York, 2011. [→3]

[100] S. M. Fosson. Non-convex approach to binary compressed sensing. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1959–1963, 2018. [→67]

[101] S. M. Fosson and M. Abuabiah. Recovery of binary sparse signals from compressed linear measurements via polynomial optimization. *IEEE Signal Processing Letters*, 26(7):1070–1074, 2019. [→67]

[102] S. Foucart and R. Gribonval. Real versus complex null space properties for sparse vector recovery. *Comptes Rendus Mathematique*, 348(15):863–865, 2010. [→78]

[103] S. Foucart and M.-J. Lai. Sparsest solutions of underdetermined linear systems via $\ell_q$-minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis*, 26(3):395–407, 2009. [→142, 143]

[104] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing.* Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013. [→2, 3, 4, 5, 6, 13, 14, 24, 25, 27, 29, 35, 40, 41, 43, 47, 64, 99, 100, 101, 102, 103, 104, 105, 106, 112, 118, 119, 142, 203, 206, 209]

[105] A. Frieze and M. Jerrum. Improved approximation algorithms for MAXk-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997. [→175]

[106] C. Fritz. Some fixed point basics. In E. Grädel, W. Thomas, and T. Wilke, editors, *Automata Logics, and Infinite Games: A Guide to Current Research*, pages 359–364. Springer Berlin Heidelberg, 2002. [→168]

[107] J.-J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004. [→45]

[108] T. Fuchs, D. Gross, P. Jung, F. Krahmer, R. Kueng, and D. Stöger. Proof methods for robust low-rank matrix recovery. Preprint, arXiv:2106.04382, 2021. [→115]

[109] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, 11(3):647–674, 2001. [→54]

[110] T. Gally. *Computational Mixed-Integer Semidefinite Programming.* PhD thesis, TU Darmstadt, 2019. [→147, 155, 158, 159, 160, 174, 175, 176, 178, 179]

[111] T. Gally and M. E. Pfetsch. Computing restricted isometry constants via mixed-integer semidefinite programming. Preprint, Optimization Online, 2016. `http://www.optimization-online.org/DB_HTML/2016/04/5395.html`. [→7, 124, 142, 144, 163, 176]

[112] T. Gally, M. E. Pfetsch, and S. Ulbrich. A framework for solving mixed-integer semidefinite programs. *Optimization Methods and Software*, 33(3): 594–632, 2017. [→155, 158, 159]

[113] G. Gamrath, T. Fischer, T. Gally, A. M. Gleixner, G. Hendel, T. Koch, S. J. Maher, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, S. Schenker, R. Schwarz, F. Serrano, Y. Shinano, S. Vigerske, D. Weninger,

M. Winkler, J. T. Witt, and J. Witzig. The SCIP Optimization Suite 3.2. ZIB-Report 15-60, Zuse Institute Berlin, 2016. `http://nbn-resolving.de/urn:nbn:de:0297-zib-57675`. [→129]

[114] G. Gamrath, D. Anderson, K. Bestuzheva, W.-K. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. Le Bodic, S. J. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M. E. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig. The SCIP Optimization Suite 7.0. ZIB-Report 20-10, Zuse Institute Berlin, 2020. `http://nbn-resolving.de/urn:nbn:de:0297-zib-78023`. [→129, 173]

[115] F. Gantmacher. *Theory of matrices*, volume 2. AMS Chelsea Publishing, 1959. [→147]

[116] D. Ge, X. Jiang, and Y. Ye. A note on the complexity of $l_p$ minimization. *Mathematical Programming*, 129(2):285–299, 2011. [→203]

[117] P. Gemander, W.-K. Chen, D. Weninger, L. Gottwald, A. Gleixner, and A. Martin. Two-row and two-column mixed-integer presolve using hashing-based pairing methods. *EURO Journal on Computational Optimization*, 8: 205–240, 2020. [→158]

[118] B. Ghaddar, M. F. Anjos, and F. Liers. A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem. *Annals of Operations Research*, 188(1):155–188, 2011. [→175]

[119] J. Gleeson and J. Ryan. Identifying minimally infeasible subsystems of inequalities. *ORSA Journal on Computing*, 2(1):61–63, 1990. [→54]

[120] A. M. Gleixner, T. Berthold, B. Müller, and S. Weltge. Three enhancements for optimization-based bound tightening. *Journal of Global Optimization*, 67 (4):731–757, 2017. [→159]

[121] Y. Gordon. On Milman's inequality and random subspaces which escape through a mesh in $\mathbb{R}^n$. In J. Lindenstrauss and V. D. Milman, editors, *Geometric Aspects of Functional Analysis*, pages 84–106, Berlin, Heidelberg, 1988. Springer. [→99, 103, 104]

[122] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.2. `http://cvxr.com/cvx`, 2014. [→130]

[123] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12):3320–3325, 2003. [→5, 14]

[124] K. Gröchenig. *Foundations of time-frequency analysis.* Applied and Numerical Harmonic Analysis. Birkhäuser, Boston, 2001. [→15]

[125] G. Hegde, Y. Yang, C. Steffens, and M. Pesavento. Parallel low-complexity M-PSK detector for large-scale MIMO systems. In *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5, 2016. [→66]

[126] G. Hegde, M. Pesavento, and M. E. Pfetsch. Joint active device identification and symbol detection using sparse constraints in massive MIMO systems. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 703–707, 2017. [→66]

[127] M. A. Herman and T. Strohmer. High-resolution radar via compressed sensing. *IEEE Transactions on Signal Processing*, 57(6):2275–2284, 2009. [→2]

[128] J. Heuer, F. Matter, M. E. Pfetsch, and T. Theobald. Block-sparse recovery of semidefinite systems and generalized null space conditions. *Linear Algebra and its Applications*, 603:470–495, 2020. [→6, 9, 14, 15, 18, 21, 30, 31, 51, 58]

[129] N. J. Higham, N. Strabić, and V. Šego. Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block. *SIAM Review*, 58(2):245–263, 2016. [→169]

[130] R. A. Horn and C. R. Johnson. *Matrix analysis.* Cambridge University Press, 2nd edition, 2012. [→10, 143]

[131] W. C. Horrace. Moments of the truncated normal distribution. *Journal of Productivity Analysis*, 43(2):133–138, 2015. [→207]

[132] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches.* Springer, Berlin, 3 edition, 1996. ISBN 3540610383. [→85]

[133] IBM ILOG. *CPLEX User's Manual Version 12 Release 1*, 2017. [→90]

[134] R. Jiang, D. Li, and B. Wu. SOCP reformulation for the generalized trust region subproblem via a canonical form of two symmetric matrices. *Mathematical Programming*, 169:531–563, 2018. [→169]

[135] A. Juditsky and A. Nemirovski. On verifiable sufficient conditions for sparse signal recovery via $\ell_1$ minimization. *Mathematical Programming*, 127(1):57–88, 2011. [→125]

[136] A. Juditsky, F. K. Karzan, and A. Nemirovski. Verifiable conditions of $\ell_1$-recovery for sparse signals with sign restrictions. *Mathematical Programming*, 127(1):89–122, 2011. [→29, 30, 40, 125]

[137] A. Juditsky, F. K. Karzan, and A. Nemirovski. On a unified view of nullspace-type conditions for recoveries associated with general sparsity structures. *Linear Algebra and its Applications*, 441:124–151, 2014. [→6, 7, 14, 15, 16, 23, 24, 58, 197]

[138] M. Kabanava and H. Rauhut. Cosparsity in compressed sensing. In H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, editors, *Compressed Sensing and its Applications: MATHEON Workshop 2013*, pages 315–339. Springer International Publishing, Cham, 2015. [→15, 23]

[139] M. Kabanava, R. Kueng, H. Rauhut, and U. Terstiege. Stable low-rank matrix recovery via null space properties. *Information and Inference: A Journal of the IMA*, 5(4):405–441, 2016. [→30, 99, 115, 116]

[140] S. Keiper. Recovery of binary sparse signals from structured biased measurements. Preprint, arXiv:2006.14835, 2020. [→67]

[141] S. Keiper, G. Kutyniok, D. G. Lee, and G. E. Pfander. Compressed sensing for finite-valued signals. *Linear Algebra and its Applications*, 532:570–613, 2017. [→51, 67, 100]

[142] K. Kellner, M. E. Pfetsch, and T. Theobald. Irreducible infeasible subsystems of semidefinite systems. *Journal of Optimization Theory and Applications*, 181 (3):727–742, 2019. [→54]

[143] M. A. Khajehnejad, A. G. Dimakis, W. Xu, and B. Hassibi. Sparse recovery of nonnegative signals with minimal expansion. *IEEE Transactions on Signal Processing*, 59(1):196–208, 2011. [→14, 25, 62]

[144] M. Kliesch, S. J. Szarek, and P. Jung. Simultaneous structures in convex signal recovery—revisiting the convex combination of norms. *Frontiers in Applied Mathematics and Statistics*, 5:1–16, 2019. [→95]

[145] K. Kobayashi and Y. Takano. A branch-and-cut algorithm for solving mixed-integer semidefinite optimization problems. *Computational Optimization and Applications*, 75(2):493–513, 2020. [→142, 155, 157, 174, 177, 179]

[146] L. Kong, J. Sun, and N. Xiu. S-semigoodness for low-rank semidefinite matrix recovery. *Pacific Journal of Optimization*, 10(1):73–83, 2014. [→14, 26, 30, 41, 62]

[147] F. Krahmer, C. Kruschel, and M. Sandbichler. Total variation minimization in compressed sensing. In H. Boche, G. Caire, R. Calderbank, M. März,

G. Kutyniok, and R. Mathar, editors, *Compressed Sensing and its Applications*, Applied and Numerical Harmonic Analysis, pages 333–358. Springer International Publishing, Cham, 2017. [→23]

[148] K. Krishnan and J. E. Mitchell. A unifying framework for several cutting plane methods for semidefinite programming. *Optimization Methods and Software*, 21(1):57–74, 2006. [→155]

[149] R. Kueng and P. Jung. Robust nonnegative sparse recovery and the nullspace property of 0/1 measurements. *IEEE Transactions on Information Theory*, 64(2):689–703, 2018. [→29]

[150] J. Kuske, P. Swoboda, and S. Petra. A novel convex relaxation for non-binary discrete tomography. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 235–246. Springer, 2017. [→66]

[151] M.-J. Lai and Y. Liu. The null space property for sparse recovery from multiple measurement vectors. *Applied and Computational Harmonic Analysis*, 30(3): 402–406, 2011. [→52]

[152] P. Lancaster and L. Rodman. Canonical forms for Hermitian matrix pairs under strict equivalence and congruence. *SIAM Review*, 47(3):407–443, 2005. [→169]

[153] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming 1958-2008*, pages 105–132. Springer Berlin Heidelberg, 2010. [→11]

[154] J.-H. Lange, M. E. Pfetsch, B. M. Seib, and A. M. Tillmann. Sparse recovery with integrality constraints. *Discrete Applied Mathematics*, 283:346–366, 2020. [→45, 48, 51, 67, 68, 70, 73, 74]

[155] A. Leshem and A.-J. van der Veen. Direction-of-arrival estimation for constant modulus signals. *IEEE Transactions on Signal Processing*, 47(11):3125–3129, 1999. [→78]

[156] C. Li and B. Adcock. Compressed sensing with local structure: Uniform recovery guarantees for the sparsity in levels class. *Applied and Computational Harmonic Analysis*, 46(3):453–477, 2019. [→94]

[157] Y. Li and W. Xie. Exact and approximation algorithms for sparse PCA. Preprint, Optimization Online, 2020. `http://www.optimization-online.org/DB_HTML/2020/05/7802.html`. [→144, 195]

[158] C. Liaw, A. Mehrabian, Y. Plan, and R. Vershynin. A simple tool for bounding the deviation of random matrices on geometric sets. In B. Klartag and E. Milman, editors, *Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 2014–2016*, pages 277–299. Springer International Publishing, Cham, 2017. [→99]

[159] J. H. Lin and S. Li. Block sparse recovery via mixed $l_2/l_1$ minimization. *Acta Mathematica Sinica, English Series*, 29(7):1401–1412, 2013. [→52]

[160] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *IEEE International Symposium on Computer Aided Control Systems Design*, pages 284–289, 2004. [→155]

[161] B. F. Logan. *Properties of high-pass signals.* PhD thesis, Columbia University, 1965. [→4]

[162] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991. [→164]

[163] Y. M. Lu and M. N. Do. A theory for sampling signals from a union of subspaces. *IEEE Transactions on Signal Processing*, 56(6):2334–2345, 2008. [→95]

[164] M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma. Polyhedral approximation in mixed-integer convex optimization. *Mathematical Programming*, 172:139–168, 2018. [→155]

[165] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010. [→164]

[166] M. Lustig, D. L. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007. [→2]

[167] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed Sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72–82, 2008. [→2]

[168] A. Mahajan. Presolving mixed–integer linear programs. In *Wiley Encyclopedia of Operations Research and Management Science.* American Cancer Society, 2011. [→158]

[169] S. J. Maher, T. Fischer, T. Gally, G. Gamrath, A. Gleixner, R. L. Gottwald, G. Hendel, T. Koch, M. E. Lübbecke, M. Miltenberger, B. Müller, M. E.

Pfetsch, C. Puchert, D. Rehfeldt, S. Schenker, R. Schwarz, F. Serrano, Y. Shinano, D. Weninger, J. T. Witt, and J. Witzig. The SCIP Optimization Suite 4.0. ZIB-Report 17-12, Zuse Institute Berlin, 2017. `http://nbn-resolving.de/urn:nbn:de:0297-zib-62170`. [→90]

[170] A. Majumdar, G. Hall, and A. A. Ahmadi. Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):331–360, 2020. [→54, 191]

[171] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008. [→15]

[172] O. Mangasarian and B. Recht. Probability of unique integer solution to a system of linear equations. *European Journal of Operational Research*, 214(1): 27–30, 2011. [→66]

[173] S. Mars. *Mixed-Integer Semidefinite Programming with an Application to Truss Topology Design*. PhD thesis, FAU Erlangen-Nürnberg, 2013. [→155, 158, 159, 160, 177, 179]

[174] F. Matter and M. E. Pfetsch. Presolving for mixed-integer semidefinite optimization. Preprint, Optimization Online, 2021. `http://www.optimization-online.org/DB_HTML/2021/10/8614.html`. [→9, 124, 142, 143, 151]

[175] F. Matter, T. Fischer, M. Pesavento, and M. E. Pfetsch. Ambiguities in Direction-of-Arrival estimation with linear arrays. Preprint, arXiv:2110.10756, 2021. [→9]

[176] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976. [→126]

[177] P. McMullen and G. C. Shephard. Diagrams for centrally symmetric polytopes. *Mathematika*, 15(2):123–138, 1968. [→64]

[178] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann. Uniform uncertainty principle for Bernoulli and subgaussian ensembles. *Constructive Approximation*, 28(3):277–289, 2008. [→99]

[179] M. Mishali and Y. C. Eldar. Blind multiband signal reconstruction: Compressed sensing for analog signals. *IEEE Transactions on Signal Processing*, 57(3):993–1009, 2009. [→53]

[180] K. Mohan and M. Fazel. New restricted isometry results for noisy low-rank recovery. In *2010 IEEE International Symposium on Information Theory*, pages 1573–1577, 2010. [→29]

[181] MOSEK ApS. *MOSEK Optimizer API for C Release 9.2.40*, 2021. URL `docs.mosek.com/9.2/capi/index.html`. [→155, 156, 173]

[182] N. Mourad and P. Reilly. Minimizing nonconvex functions for sparse vector reconstruction. *IEEE Transactions on Signal Processing*, 58(7):3485–3496, 2010. [→203]

[183] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results. *Mathematical Programming*, 95(2):303–327, 2003. [→54]

[184] S. Nam, M. Davies, M. Elad, and R. Gribonval. The cosparse analysis model and algorithms. *Applied and Computational Harmonic Analysis*, 34(1):30–56, 2013. [→15]

[185] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995. [→4, 13]

[186] D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26 (3):301–321, 2009. [→4]

[187] D. Needell and R. Ward. Stable image reconstruction using total variation minimization. *SIAM Journal on Imaging Sciences*, 6(2):1035–1058, 2013. [→15]

[188] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of $M$-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012. [→6, 19, 200]

[189] C. J. Nohra, A. U. Raghunathan, and N. Sahinidis. Spectral relaxations and branching strategies for global optimization of mixed-integer quadratic programs. *SIAM Journal on Optimization*, 31(1):142–171, 2021. [→163]

[190] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Mathematical Programming*, 62(1):321–357, 1993. [→140]

[191] D. B. Owen. A table of normal integrals. *Communications in Statistics – Simulation and Computation*, 9(4):389–419, 1980. [→211]

[192] S. Oymak and B. Hassibi. New null space results and recovery thresholds for matrix rank minimization. Preprint, arXiv:1011.6326, 2010. In Proceedings of ISIT 2011. [→6, 14, 25, 26, 45, 48, 62, 100]

[193] S. Oymak and B. Hassibi. Tight recovery thresholds and robustness analysis for nuclear norm minimization. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 2323–2327, 2011. [→30, 99]

[194] S. Oymak, M. A. Khajehnejad, and B. Hassibi. Improved thresholds for rank minimization. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5988–5991, 2011. [→45, 48]

[195] S. Oymak, K. Mohan, M. Fazel, and B. Hassibi. A simplified approach to recovery conditions for low rank matrices. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 2318–2322, 2011. [→30, 99]

[196] S. Oymak, A. Jalali, M. Fazel, Y. C. Eldar, and B. Hassibi. Simultaneously structured models with application to sparse and low-rank matrices. *IEEE Transactions on Information Theory*, 61(5):2886–2908, 2015. [→95]

[197] F. Parvaresh, H. Vikalo, S. Misra, and B. Hassibi. Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):275–285, 2008. [→53]

[198] F. Permenter and P. A. Parrilo. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. *Mathematical Programming*, 171 (1):1–54, 2018. [→157]

[199] F. Permenter and P. A. Parrilo. Dimension reduction for semidefinite programs via Jordan algebras. *Mathematical Programming*, 181(1):51–84, 2020. [→157]

[200] F. Permenter, H. A. Friberg, and E. D. Andersen. Solving conic optimization problems via self-dual embedding and facial reduction: A unified approach. *SIAM Journal on Optimization*, 27(3):1257–1282, 2017. [→157]

[201] M. Pilanci, M. J. Wainwright, and L. El Ghaoui. Sparse learning via Boolean relaxations. *Mathematical Programming*, 151(1):62–87, 2015. [→175]

[202] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies. Sparse representations in audio and music: From coding to source separation. *Proceedings of the IEEE*, 98(6):995–1005, 2010. [→2]

[203] T. K. Pong and H. Wolkowicz. The generalized trust region subproblem. *Computational Optimization and Applications*, 58:273–322, 2014. [→169]

[204] L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin. Sparsity and compressed sensing in radar imaging. *Proceedings of the IEEE*, 98(6):1006–1020, 2010. [→2]

[205] Y. Puranik and N. V. Sahinidis. Domain reduction techniques for global NLP and MINLP optimization. *Constraints*, 22(3):338–376, 2017. [→152, 158]

[206] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58(1):353–367, 1993. [→171]

[207] A. Qualizza, P. Belotti, and F. Margot. Linear programming relaxations of quadratically constrained quadratic programs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 407–426. Springer New York, 2012. [→148]

[208] M. Ramana and A. J. Goldman. Some geometric results in semidefinite programming. *Journal of Global Optimization*, 7(1):33–50, 1995. [→121]

[209] B. Recht, W. Xu, and B. Hassibi. Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. In *2008 47th IEEE Conference on Decision and Control*, pages 3065–3070, 2008. [→25, 62]

[210] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3): 471–501, 2010. [→6, 14, 29, 140]

[211] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970. [→117]

[212] A. Ron and Z. Shen. Affine systems in $l_2(\mathbb{R}^d)$: The analysis of the analysis operator. *Journal of Functional Analysis*, 148(2):408–447, 1997. [→15]

[213] S. M. Ross. *Introduction to probability models.* Academic Press, San Diego, 12th edition, 2019. [→10, 101]

[214] V. Roulet, N. Boumal, and A. d'Aspremont. Computational complexity versus statistical performance on sparse recovery problems. *Information and Inference: A Journal of the IMA*, 9(1):1–32, 2019. [→19, 20, 23, 200]

[215] M. Rudelson and R. Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008. [→46, 99, 104, 106, 116]

[216] J. Saunderson, P. A. Parrilo, and A. S. Willsky. Semidefinite descriptions of the convex hull of rotation matrices. *SIAM Journal on Optimization*, 25(3): 1314–1343, 2015. [→121]

[217] M. W. P. Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing*, 6(4):445–454, 1994. [→158, 159]

[218] A. Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998. [→10, 86]

[219] SCIP. Solving Constraint Integer Programs. `http://scip.zib.de`. [→129, 151]

[220] SCIP-SDP. a mixed-integer semidefinite programming plugin for SCIP. `http://www.opt.tu-darmstadt.de/scipsdp/`. [→7, 151, 155]

[221] I. W. Selesnick and M. A. T. Figueiredo. Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors. In V. K. Goyal, M. Papadakis, and D. V. D. Ville, editors, *Wavelets XIII*, volume 7446, pages 107–121. International Society for Optics and Photonics, 2009. [→15]

[222] Y. Shechtman, A. Beck, and Y. C. Eldar. GESPAR: Efficient phase retrieval of sparse signals. *IEEE Transactions on Signal Processing*, 62(4):928–938, 2014. [→90]

[223] H. D. Sherali and B. M. Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22:233–261, 2002. [→155]

[224] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013. [→94]

[225] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. C. Eldar. C-HiLasso: A collaborative hierarchical sparse modeling framework. *IEEE Transactions on Signal Processing*, 59(9):4183–4198, 2011. [→94]

[226] M. Stojnic. Various thresholds for $\ell_1$-optimization in compressed sensing. Preprint, arXiv:0907.3666, 2009. [→45, 48, 99, 101, 104, 121]

[227] M. Stojnic. Recovery thresholds for $\ell_1$ optimization in binary compressed sensing. In *2010 IEEE International Symposium on Information Theory*, pages 1593–1597. IEEE, 2010. [→66]

[228] M. Stojnic. $\ell_2/\ell_1$-optimization in block-sparse compressed sensing and its strong thresholds. *IEEE Journal of Selected Topics in Signal Processing*, 4(2): 350–357, 2010. [→116]

[229] M. Stojnic. Compressed sensing of block-sparse positive vectors. Preprint, arXiv:1306.3977, 2013. [→14, 52, 99, 101, 121]

[230] M. Stojnic, F. Parvaresh, and B. Hassibi. On the reconstruction of block-sparse signals with an optimal number of measurements. *IEEE Transactions on Signal Processing*, 57(8):3075–3085, 2009. [→6, 14, 52, 61, 62, 99, 121]

[231] N. Strabić. *Theory and algorithms for matrix problems with positive semidefinite constraints.* PhD thesis, University of Manchester, 2016. [→169]

[232] C. Studer, T. Goldstein, W. Yin, and R. G. Baraniuk. Democratic representations. Preprint, arXiv:1401.3420, 2014. [→90]

[233] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht. Compressed sensing off the grid. *IEEE Transactions on Information Theory*, 59(11):7465–7490, 2013. [→100]

[234] A. Tarski. A lattice-theoretical fixpoint theorem and its application. *Pacific Journal of Mathematics*, 5:285–309, 1955. [→168]

[235] A. M. Tillmann. *Computational aspects of compressed sensing.* PhD thesis, TU Darmstadt, 2013. [→3]

[236] A. M. Tillmann. Computing the spark: mixed-integer programming for the (vector) matroid girth problem. *Computational Optimization and Applications*, 74(2):387–441, 2019. [→125]

[237] A. M. Tillmann and M. E. Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2014. [→24, 125, 142, 202]

[238] Y. Traonmilin and R. Gribonval. Stable recovery of low-dimensional cones in hilbert spaces: One RIP to rule them all. *Applied and Computational Harmonic Analysis*, 45(1):170–205, 2018. [→200]

[239] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004. [→4, 5]

[240] J. Tropp. Recovery of short, complex linear combinations via $\ell_1$ minimization. *IEEE Transactions on Information Theory*, 51(4):1568–1570, 2005. [→45]

[241] J. A. Tropp. Convex recovery of a structured signal from independent random linear measurements. In G. E. Pfander, editor, *Sampling Theory, a Renaissance: Compressive Sensing and Other Developments*, pages 67–101. Springer International Publishing, Cham, 2015. [→99, 103]

[242] E. van den Berg and M. P. Friedlander. Theoretical and empirical results for recovery from multiple measurements. *IEEE Transactions on Information Theory*, 56(5):2516–2527, 2010. [→52]

[243] A.-J. van der Veen and A. Paulraj. An analytical constant modulus algorithm. *IEEE Transactions on Signal Processing*, 44(5):1136–1155, 1996. [→78]

[244] L. Vandenberghe and M. S. Andersen. Chordal graphs and semidefinite optimization. *Foundations and Trends in Optimization*, 1(4):241–433, 2015. [→54]

[245] A. M. Vershik and P. V. Sporyshev. Asymptotic behavior of the number of faces of random polyhedra and the neighborliness problem. *Selecta Mathematica Sovietica*, 11(2):181–201, 1992. [→100]

[246] R. Vershynin. Estimation in high dimensions: A geometric perspective. In G. E. Pfander, editor, *Sampling Theory, a Renaissance: Compressive Sensing and Other Developments*, pages 3–66. Springer International Publishing, Cham, 2015. [→100]

[247] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018. [→100]

[248] M. Vidyasagar. *An introduction to compressed sensing*. SIAM, 2019. [→3, 97, 100]

[249] S. Vigerske. *Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming.* PhD thesis, Humboldt-Universität zu Berlin, 2013. [→87, 158]

[250] S. Vigerske and A. Gleixner. SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, 33(3):563–593, 2018. [→85, 158]

[251] J. Witzig. *Infeasibility Analysis for MIP*. PhD thesis, TU Berlin, 2021. [→159]

[252] J. Witzig, T. Berthold, and S. Heinz. Experiments with conflict analysis in mixed integer programming. In *Integration of AI and OR Techniques in Constraint Programming. CPAIOR 2017*, volume 10335, pages 211–222, 2017. [→159]

[253] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. [→53]

[254] M. Yamashita, K. Fujisawa, M. Fukuda, K. Nakata, and M. Nakata. A high-performance software package for semidefinite programs: SDPA 7. Research Report B-460, Department of Mathematical and Computing Science, Tokyo Institute of Technology, Tokyo, Japan, 2010. [→155]

[255] X.-T. Yuan and T. Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14(4):899–925, 2013. [→148]

[256] Y. Zhang. A simple proof for recoverability of $\ell_1$-minimization (II): The non-negativity case. Technical report TR05-10, Dept. of Computational and Applied Mathematics, Rice University, 2005. [→14, 25, 62]

[257] G. Ziegler. *Lectures on Polytopes.* Graduate Texts in Mathematics. Springer, New York, 1995. [→64]

[258] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006. [→143]

# List of Algorithms

# List of Figures

# List of Tables

# Wissenschaftlicher Werdegang

07/2017 – 09/2022 Wissenschaftlicher Mitarbeiter am Fachbereich Mathematik der Technischen Universität Darmstadt in der Arbeitsgruppe Diskrete Optimierung und im Schwerpunktprogramm 1798 „Compressed Sensing in der Informationsverarbeitung (CoSIP)“

05/2017 Abschluss Master of Science in Mathematik

10/2011 – 05/2017 Studium der Mathematik an der Goethe-Universität Frankfurt am Main

06/2011 Abitur am Franziskanergymnasium Kreuzburg in Großkrotzenburg