Lösung großer konischer Programme mit Hilfe primal-dualer Methoden

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Thomas Davi

aus Groß Strehlitz

Düsseldorf, Mai 2012

aus dem Institut für Mathematische Optimierung
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
der Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Florian Jarre
Korreferent: Prof. Dr. Franz Rendl

Tag der mündlichen Prüfung: 5.7.2012

Zusammenfassung

Sei ein reeller endlichdimensionaler Hilbertraum $(E, \langle ., . \rangle)$ gegeben. Wir betrachten ein konisches Programm von der Form

$$(P) \qquad \text{minimiere} \quad \langle c, x \rangle \mid x \in (\mathcal{L} + b) \cap \mathcal{K}.$$

Hierbei seien $\mathcal{L} \subseteq E$ ein Unterraum, $\mathcal{K} \subseteq E$ ein nicht-leerer, konvexer und abgeschlossener Kegel und $b, c \in E$.

Eine Vielzahl von Problemstellungen in Wirtschaft und Wissenschaft kann durch konische Programme beschrieben werden.

Wichtige Spezialfälle sind lineare (mit $E = I\!R^n$ und $\mathcal{K} = I\!R^n_+$) und semidefinite Programme (mit $E = \mathcal{S}^n$ und $\mathcal{K} = \mathcal{S}^n_+$). Für beide Klassen existieren viele effiziente Lösungsmethoden, solange die Dimension der Probleme nicht zu groß wird. Eine schwieriger zu handhabende Klasse sind die doppelt nichtnegativen Programme (mit $E = \mathcal{S}^n$ und $\mathcal{K} = \mathcal{S}^n_+ \cap \{X \geq 0\}$). Mit ihnen werden u.a. Probleme in der Graphentheorie gelöst. Im Gegensatz zu linearen und semidefiniten Programmen ist der Kegel in diesem Fall nicht selbstdual.

Zunächst widmet sich die Arbeit allgemeinen konischen Programmen der Form (P) und den zugehörigen dualen Programmen (D). Es wird eine Umformulierung der beiden Programme betrachtet, die es ermöglicht alle Lösungen des Paares (P) und (D) als Schnittmenge eines affinen Raumes $\mathbf{A}$ und eines Kegels $\mathbf{K}$ im Raum $E \times E$ darzustellen. Anschließend wird die APD-Methode zur Lösung von konischen Programmen besprochen. Im Wesentlichen wird bei dieser Methode ausgehend von $\mathbf{A}$ der Abstand zum Kegel $\mathbf{K}$ minimiert.

Der Spezialfall der semidefiniten Programme wird genauer untersucht: Die APD-Methode wird für diese Problemklasse analysiert. Weiterhin wird auf eine Regularisierung eingegangen, welche das Konvergenzverhalten des APD-Verfahrens verbessern soll. Schließlich werden hochdimensionale Programme in Bezug auf Speicherbedarf und Rechenaufwand besprochen.

Es werden zwei Erweiterungen des APD-Verfahrens betrachtet: Zunächst wird das verallgemeinerte Newton-Verfahren diskutiert. Danach wird das AHO-QMR-Verfahren beschrieben. Dabei handelt es sich um ein iteratives Verfahren, welches mit Hilfe der QMR-Methode die aus den Innere-Punkte-Methoden bekannte AHO-Richtung approximiert. Hierbei wird das zu lösende Gleichungssystem zunächst symmetrisiert. Neben einer ausführlichen Beschreibung der notwendigen Umformulierungsschritte werden auch numerische Ergebnisse präsentiert.

Schließlich wird auf den doppelt nichtnegativen Kegel und die zugehörigen konischen Programme eingegangen. Es wird zunächst eine äquivalente selbstduale Formulierung angegeben und untersucht, wie sich die APD- und die AHO-QMR-Methode auf diese Formulierung anwenden lassen. Anschließend werden Regularitätsbedingungen für das Ausgangsproblem betrachtet, die die Konvergenz der APD- und AHO-QMR-Methode zur Lösung der selbstudalen Programme verbessern. Abschließend werden numerische Ergebnisse angegeben.

Abstract

Let $(E, \langle ., . \rangle)$ be a real finite-dimensional Hilbert space. Consider a conic program of the form

$$(P) \qquad \text{minimize} \quad \langle c, x \rangle \mid x \in (\mathcal{L} + b) \cap \mathcal{K}.$$

Here, $\mathcal{L} \subseteq E$ is a linear subspace, $\mathcal{K} \subseteq E$ a non-empty, convex and closed cone and $b, c \in E$ given data.

A broad variety of economic and scientific problems can be described by conic programs.

Important special cases are linear ($E = I\!R^n$ and $\mathcal{K} = I\!R^n_+$) and semidefinite programs ($E = \mathcal{S}^n$ and $\mathcal{K} = \mathcal{S}^n_+$). For both cases, when the problem size is not "too large", there are many efficient problem-solving approaches. A somewhat more difficult case are doubly nonnegative programs ($E = \mathcal{S}^n$ and $\mathcal{K} = \mathcal{S}^n_+ \cap \{X \geq 0\}$). They are used for solving problems in graph theory, for example. In contrary to linear and semidefinite programs, the cone is not selfdual in this case.

First, the thesis focuses on general conic programs of the form (P) and the corresponding dual programs (D). A reformulation of both programs, which allows to represent the solution of the pair (P) and (D) as the intersection of an affine space $\mathbf{A}$ and a cone $\mathbf{K}$ in $E \times E$, is considered. Next, the APD-Method for solving conic programs is discussed. Basically, starting from $\mathbf{A}$ the distance to $\mathbf{K}$ is minimized in this method.

The special case of semidefinite programs is investigated. The APD-Method is analyzed for this problem class. Furthermore, a regularization to speed up convergence of the APD-Method is discussed. Finally, high dimensional programs are considered.

Following, two extensions of the APD-Method are examined: An application of the generalized Newton-Method and the AHO-QMR-Method. The latter one is an iterative method, which uses the QMR-algorithm to approximate the AHO-direction known from interior point methods. Here, the system of equations to be solved is symmetrized at first. A detailed description of all necessary reformulation steps and numerical results are reported.

Finally, the thesis focuses on the doubly nonnegative cone and the corresponding conic programs. First, an equivalent selfdual formulation is described. Then, a generalization of the APD- and AHO-QMR-Method for this formulation is considered. Furthermore, regularization conditions for the initial problem are presented, which guarantee an improved convergence of the APD- and AHO-QMR-Method for solving the selfdual formulation. Concluding, several numerical results are stated.

Inhaltsverzeichnis

Einleitung

Die vorliegende Arbeit beschäftigt sich mit der Lösung konischer Programme. Sei zunächst ein reeller endlichdimensionaler Hilbertraum $(E, \langle ., . \rangle)$ gegeben. Ein konisches Programm hat die Form

$$(P) \qquad \text{minimiere} \quad \langle c, x \rangle \mid x \in (\mathcal{L} + b) \cap \mathcal{K}.$$

Hierbei seien $\mathcal{L} \subseteq E$ ein Unterraum, $\mathcal{K} \subseteq E$ ein nicht-leerer, konvexer und abgeschlossener Kegel und b, c Elemente aus E. Eine Vielzahl von Problemstellungen in Wirtschaft und Wissenschaft kann durch konische Programme beschrieben werden.

Ein wichtiger Spezialfall sind die linearen Programme, mit deren Hilfe beispielsweise Flughafenpläne oder Versandwege und Kosten eines Versandhauses mathematisch formuliert und mit geeigneten Methoden optimiert werden können. Zur Lösung von linearen Programmen existieren bereits viele effiziente Algorithmen.

Eine weitere wichtige Klasse von konischen Programmen sind die semidefiniten Programme. Hiermit können u.a. Probleme aus dem Bereich der Automatisierungstechnik formuliert werden. Auch für diese Klasse ist eine ganze Reihe von Lösungsmethoden bekannt. Hat man es bei linearen oder semidefiniten Programmen jedoch mit einer hohen Anzahl an Unbekannten zu tun, dann können viele der sehr genauen Verfahren nicht mehr verwendet werden, da die oftmals benötigte Berechnung und Speicherung von Informationen „zweiter Ordnung" zu teuer ist.

Eine schwieriger zu handhabende Klasse sind die doppelt nichtnegativen Programme. Sie dienen u.a. dazu Lösungen von quadratischen Problemen anzunähern, mit welchen wiederum viele geometrische Problemstellungen gelöst werden. Weiterhin lassen sich Lösungen von Optimierungsaufgaben wie dem Max-Cut- oder Max-Clique-Problem sowohl durch semidefinite als auch durch doppelt nichtnegative Programme annähern. Die letztgenannten Prgramme liefern dabei oftmals die „besseren" Approximationen der gesuchten Lösung.

Das erste Kapitel der Arbeit widmet sich allgemeinen konischen Programmen der Form (P). Dabei wird zuerst auf die Dualitätstheorie eingegangen, in der zu jedem primalen Programm (P) ein duales Programm (D) betrachtet wird. Anschließend wird eine Umformulierung der beiden Programme betrachtet, die es

ermöglicht alle Lösungen des Paares (P) und (D) als Schnittmenge eines affinen Raumes $\mathbf{A}$ und eines Kegels $\mathbf{K}$ im Raum $E \times E$ darzustellen. Im Anschluss wird die APD-Methode zur Lösung von konischen Programmen besprochen. Sie wurde zuerst in [9] vorgeschlagen. Im Wesentlichen wird bei der APD-Methode ausgehend von $\mathbf{A}$ der Abstand zum Kegel $\mathbf{K}$ minimiert. Dabei wird der Abstand mittels einer konvexen differenzierbaren Funktion ϕ formuliert. Jedes Minimum von ϕ entspricht einer Lösung der Programme (P) und (D).

Die einfache APD-Methode ist ein iteratives Verfahren erster Ordnung – es werden nur Funktions- und Ableitungsauswertungen vorgenommen. Sie ist in allen oben betrachteten Spezialfällen selbst für hochdimensionale Programme verwendbar, sofern die zum Problem gehörende Datenmenge verhältnismäßig klein ist.

Das zweite Kapitel beschäftigt sich mit semidefiniten Programmen. Die im ersten Kapitel beschriebene Lösungsmethode wird für diese Problemklasse analysiert. Kernpunkt dabei ist die Verwendbarkeit des verallgemeinerten Newton-Verfahrens (siehe [20]) zur Minimierung von ϕ. Von besonderem Interesse ist daher der Definitionsbereich und die Form der zweiten Ableitung von ϕ. Weiterhin wird auf eine Regularisierung eingegangen, welche das Konvergenzverhalten des APD-Verfahrens bei Verwendung des Newton- oder Quasi-Newton-Verfahrens zur Suchrichtungsbestimmung in der Nähe der Optimallösung verbessern soll. Schließlich werden hochdimensionale Programme in Bezug auf Speicherbedarf, Rechenaufwand und mögliche Reduzierungen derselben besprochen und numerische Ergebnisse einer Implementierung des APD-Verfahrens geliefert.

Im dritten Kapitel geht es um Erweiterungen des APD-Verfahrens: Zunächst wird das verallgemeinerte Newton-Verfahren diskutiert. Da eine direkte Implementierung wegen des hohen Speicherbedarfs vermieden werden muss, wird eine Approximation des Newton-Verfahrens mit Hilfe eines iterativen Verfahrens (cg-Variante von Steihaug) beschrieben. Das cg-Verfahren benötigt nur Richtungsableitungen von $\nabla\phi$, welche außerhalb einer Nullmenge bestimmt werden können und deren Berechnungskosten im annehmbaren Bereich liegen. Danach wird das AHO-QMR Verfahren beschrieben. Dabei handelt es sich um ein iteratives Verfahren, welches mit Hilfe der QMR-Methode die aus den Innere-Punkte-Methoden bekannte AHO-Richtung approximiert. Hierbei wird das zu lösende Gleichungssystem durch eine Transformation des Systems und Eliminierung einiger Variablen symmetrisiert. Dadurch wird im Vergleich zur Lösung des Ausgangssystems Speicherplatz und vor allem Rechenzeit verringert, da die symmetrische QMR-Variante nicht einmal halb so viele Auswertungen benötigt. Neben einer ausführlichen Beschreibung der notwendigen Umformulierungsschritte werden auch numerische Ergebnisse präsentiert.

Im vierten Kapitel geht es schließlich um den doppelt nichtnegativen Kegel und die zugehörigen konischen Programme. Da der Kegel solcher Programme im Gegensatz zu linearen und semidefiniten Programmen nicht selbstdual ist, wird zunächst eine äquivalente selbstduale Formulierung angegeben, auf welche sich

die APD- und die AHO-QMR-Methode relativ einfach anwenden lassen. Es wird gezeigt, dass sich die Rechenzeit dieser Methoden gegenüber derjenigen für einfache semidefinite Programme nur geringfügig ändert. Anschließend werden Regularitätsbedingungen für das Ausgangsproblem angegeben, die die Konvergenzeigenschaften der APD- und AHO-QMR-Methode zur Lösung der selbstudalen Programme verbessern. Abschließend werden numerische Ergebnisse präsentiert und eine Erweiterung der beiden Methoden auf allgemeinere semidefinite Programme beschrieben, deren Spezialfälle u.a. lineare Programme, einfache semidefinite Programme und die in diesem Kapitel beschriebene selbstduale Formulierung der doppelt nichtnegativen Programme sind.

Im Zuge dieser Arbeit sind zwei gemeinsame Paper entstanden:
Das AHO-QMR-Verfahren aus Kapitel 3 ist Gegenstand der Publikation [3]. In [4] werden die in Kapitel 4 beschriebenen doppelt nichtnegativen Programme betrachtet.

Mein besonderer Dank geht an meinen Betreuer Professor Florian Jarre, welcher mir stets mit Rat zur Seite stand und die Entstehung dieser Arbeit überhaupt erst möglich machte. Außerdem danke ich meinem Mitarbeiter Li Luo für die interessanten Gespräche, die mich auf die eine oder andere nützliche Idee gebracht haben.

Kapitel 1

Konische Programme

Die Verallgemeinerung der linearen Programme von der Form

$$\min\{c^T x \mid Ax = b,\ x \geq 0\}$$

mit $c \in \mathbb{R}^n$, $b \in \mathbb{R}^d$ und $A \in \mathbb{R}^{d \times n}$ sind die im Folgenden beschriebenen konischen Programme.

Gegeben sei ein endlich-dimensionaler vollständiger $\mathbb{R}$-Vektorraum E versehen mit dem Skalarprodukt $\langle .,. \rangle := \langle .,. \rangle_E$. Wir werden Räume mit diesen Eigenschaften ab jetzt als euklidische Räume bezeichnen. Seien weiterhin Elemente $b, c \in E$, ein Unterraum $\mathcal{L} \subseteq E$ und ein nicht-leerer abgeschlossener konvexer Kegel $\mathcal{K} \subseteq E$ gegeben. Das allgemeine konische Programm hat dann die Form

$$(P) \quad \min\{\langle c, x \rangle \mid x \in (\mathcal{L} + b) \cap \mathcal{K}\}.$$

1.1 Dualität

Das zu (P) duale Problem wird in [15] in der Form

$$(D) \quad \min\{\langle b, s \rangle \mid s \in (\mathcal{L}^\perp + c) \cap \mathcal{K}^D\}$$

definiert. Dabei ist $\mathcal{L}^\perp := \{z \in E \mid \langle z, x \rangle = 0\ \forall x \in \mathcal{L}\}$ der bezüglich $\langle .,. \rangle$ zu $\mathcal{L}$ orthogonale Unterraum und $\mathcal{K}^D := \{z \in E \mid \langle z, x \rangle \geq 0\ \forall x \in \mathcal{K}\}$ der zu $\mathcal{K}$ duale Kegel. Um eine weitere Darstellung des primal-dualen Paares anzugeben, müssen wir $\mathcal{L}$ genauer beschreiben: Es gelte $\mathcal{L} = \{x \in E \mid \mathcal{A}(x) = 0\}$, wobei $\mathcal{A} : E \to \mathbb{R}^d$, $d \leq \dim(E)$, ein linearer Operator sei. Dann lässt sich (P) in der Form

$$(\bar{P}) \quad \min\{\langle c, x \rangle \mid \mathcal{A}(x) = \bar{b}, x \in \mathcal{K}\}$$

schreiben, wobei $\bar{b} = \mathcal{A}(b)$ gelte. Wie wir im Folgenden sehen werden, kann das dulae Problem dann als

$$(\bar{D}) \quad \max\{\bar{b}^T y \mid \mathcal{A}^*(y) + s = c, s \in \mathcal{K}^D\}$$

formuliert werden, wobei $\mathcal{A}^*$ der zu $\mathcal{A}$ adjungierte Operator ist. Er efüllt damit die Bedingung $y^T\mathcal{A}(z) = \langle \mathcal{A}^*(y), z\rangle_E$ für alle $y \in I\!R^d$, $z \in E$.

Aus der Beschreibung von $\mathcal{L}$ folgt zunächst $\mathcal{L}^\perp = \{\mathcal{A}^*(y) \mid y \in I\!R^d\}$. Es gilt nun: $x \in \mathcal{L} + b$ genau dann, wenn $\mathcal{A}(x) = \mathcal{A}(b) = \bar{b}$. Somit ist x genau dann zulässig für (P), wenn es für $(\bar{P})$ zulässig ist. Weiterhin ist $s \in \mathcal{L}^\perp + c$ genau dann, wenn ein $y_s \in I\!R^d$ existiert, so dass $s = c - \mathcal{A}^*(y_s)$ gilt. Damit ist s zulässig für (D) genau dann, wenn (y_s, s) zulässig für $(\bar{D})$ ist. y_s ist dabei genau dann eindeutig, wenn der Operator $\mathcal{A}^*$ injektiv ist. Weiterhin gilt für $x \in (\mathcal{L} + b) \cap \mathcal{K}$, $s \in (\mathcal{L}^\perp + c) \cap \mathcal{K}^D$

$$\begin{aligned}
\langle c, x\rangle + \langle b, s\rangle \quad &= \langle c, x\rangle + \langle b, c - \mathcal{A}^*(y_s)\rangle \\
&= \langle c, x\rangle + \langle b, c\rangle - \underbrace{\mathcal{A}(b)^T}_{=\mathcal{A}(x)} y_s \\
&= \langle c, x\rangle + \langle b, c\rangle - \langle x, \mathcal{A}^*(y_s)\rangle \qquad (1.1) \\
&= \langle c, x\rangle + \langle b, c\rangle - \langle x, c - s\rangle \\
&= \langle b, c\rangle + \underbrace{\langle x, s\rangle}_{\geq 0} \geq \langle b, c\rangle
\end{aligned}$$

Wir schließen aus Ungleichung (1.1)

$$\begin{aligned}
& \langle c, x\rangle + \langle b, s\rangle \geq \langle b, c\rangle \\
\Leftrightarrow \quad & \langle c, x\rangle + \langle b, c\rangle - \mathcal{A}(b)^T y_s \geq \langle b, c\rangle \\
\Leftrightarrow \quad & \langle c, x\rangle - \bar{b}^T y_s \geq 0 \\
\Leftrightarrow \quad & \langle c, x\rangle \geq \bar{b}^T y_s \qquad (1.2)
\end{aligned}$$

Die Ungleichungen (1.1) und (1.2) werden als *Schwache Dualität* bezeichnet.
Aus den obigen Umformulierungen wird der Zusammenhang der Formulierungen (P),(D) und $(\bar{P})$, $(\bar{D})$ klar. In der Form $(\bar{P})$ und $(\bar{D})$ ist die Ähnlichkeit zu linearen Programmen deutlich.

Die Bestimmung der Lösung konischer Programme ist mit der Aufstellung von allgemeinen Optimalitätsbedingungen verbunden, deren Nützlichkeit aber von dem speziellen Problem und insbesondere von der Regularität der betrachteten Menge abhängt. Wir wollen uns mit dieser Problematik im Folgenden auseinandersetzen. Dazu definieren wir zunächst:

Definition 1. *Sei $S \subseteq E$ eine belieibige Menge. Mit* $\mathrm{aff}(S)$ *bezeichnen wir die affine Hülle von S. Sie ist die kleinste affine Menge, die S enthält. D.h.*

$$\mathrm{aff}(S) = \bigcap_{\substack{M : M \supseteq S \\ M \ affin}} M.$$

Analog wird die konvexe Hülle $conv(S)$ definiert:

$$conv(S) = \bigcap_{\substack{M : M \supseteq S \\ M \ konvex}} M.$$

Wie üblich sei für $x \in E$ die euklidische Norm durch $\|x\|_2 = \sqrt{\langle x, x\rangle_E}$ definiert.

Definition 2. *Sei eine konvexe Menge $C \subseteq E$ gegeben. Ein Punkt $x \in \mathrm{aff}(C)$ heißt relativ innerer Punkt von C, falls es ein $\varepsilon > 0$ gibt, so dass für $B(x, \varepsilon) := \{z \in E \mid \|z - x\|_2 < \varepsilon\}$*

$$B(x, \varepsilon) \cap \mathrm{aff}(C) \subseteq C$$

gilt. Wir setzen $C^i := \{x \in E \mid x \text{ ist relativ innerer Punkt von } C\}$.

Definition 3. *Sei ein Problem der Form (P) gegeben. Wir sagen, dass (P) die Slater-Bedingung erfüllt, falls*

$$\mathcal{K}^i \cap (\mathcal{L} + b) \neq \emptyset.$$

Die Slater-Bedingung ist eine wichtige Regularitätsbedingung in der Optimierung. Da wir uns in dieser Arbeit auf konische Programme beschränken, verzichten wir auf die allgemeine Formulierung dieser Bedingung.
Einige wichtige Eigenschaften eines primal-dualen Paares (P) und (D) werden im folgenden Satz festgehalten:

Satz 1. *Für die Programme (P) und (D) gelten folgende Aussagen:*

1. *Für jeden zulässigen Punkt x von (P) und s von (D) gilt*

$$\langle c, x\rangle + \langle b, s\rangle \geq \langle b, c\rangle.$$

2. *Falls (P) einen endlichen Optimalwert α besitzt,*

$$\alpha = \inf\{\langle c, x\rangle \mid x \in \mathcal{K} \cap (\mathcal{L} + b)\} \in \mathbb{R},$$

und (P) die Slater-Bedingung erfüllt, dann besitzt (D) eine Optimallösung s^{opt} und es gilt

$$\alpha + \langle b, s^{opt}\rangle = \langle b, c\rangle.$$

Für jede Optimallösung x^{opt} von (P) gilt dann

$$\langle c, x^{opt}\rangle + \langle b, s^{opt}\rangle = \langle b, c\rangle.$$

Beweis. Teil 1 wurde mit (1.1) bereits gezeigt. Der Beweis des zweiten Teils findet sich in [10].

Da $\mathcal{K}$ ein nicht-leerer abgeschlossener konvexer Kegel ist, folgt $(\mathcal{K}^D)^D = \mathcal{K}$. Damit ist (P) das duale Programm zu (D). Wir schließen:

Korollar 1. *Falls beide Programme (P) und (D) die Slater-Bedingung erfüllen, dann besitzen sie auch Optimallösungen x^{opt} und s^{opt} mit*

$$\langle c, x^{opt}\rangle + \langle b, s^{opt}\rangle = \langle b, c\rangle.$$

Ist umgekehrt x zulässig für (P) und s zulässig für (D) und gilt

$$\langle c, x\rangle + \langle b, s\rangle = \langle b, c\rangle, \tag{1.3}$$

dann ist (x, s) Optimallösung von (P) und (D).

Somit muss (1.3) von einer Optimallösung notwendigerweise erfüllt werden, falls (P) und (D) die Slater-Bedingung erfüllen, (1.3) ist aber in jedem Falle hinreichend.

Analog zu (1.2) zeigt man

$$\langle c, x\rangle + \langle b, s\rangle = \langle b, c\rangle \Leftrightarrow \langle c, x\rangle = \bar{b}^T y_s$$

und somit folgt

(x, s) ist Optimallösung von $(P), (D) \Leftrightarrow (x, y_s, s)$ ist Optimallösung von $(\bar{P}), (\bar{D})$.

Wichtige Spezialfälle konischer Programme sind die eingangs erwähnten linearen Programme – hier gilt $E = I\!R^n$ und $\mathcal{K} = I\!R^n_+$ – sowie semidefinite Programme. Mit Letzteren werden wir uns in Kapitel 2 ausführlich beschäftigen.

1.2 Reformulierung

Sei

$$\mathbf{A} := (\mathcal{L} + b) \times (\mathcal{L}^\perp + c) \cap \{(x, s) \mid \langle c, x\rangle + \langle b, s\rangle = \langle b, c\rangle\} \subseteq E \times E$$

und

$$\mathbf{K} := \mathcal{K} \times \mathcal{K}^D \subseteq E \times E.$$

Bei den folgenden zunächst allgemeinen Überlegungen nehmen wir an, dass die folgende Voraussetzung erfüllt ist:

Voraussetzung 1. $\mathbf{A} \cap \mathbf{K} \neq \emptyset$.

Die folgende Voraussetzung ist stärker:

Voraussetzung 2. *Die zueinander dualen Programme (P) und (D) erfüllen die Slater-Bedingung.*

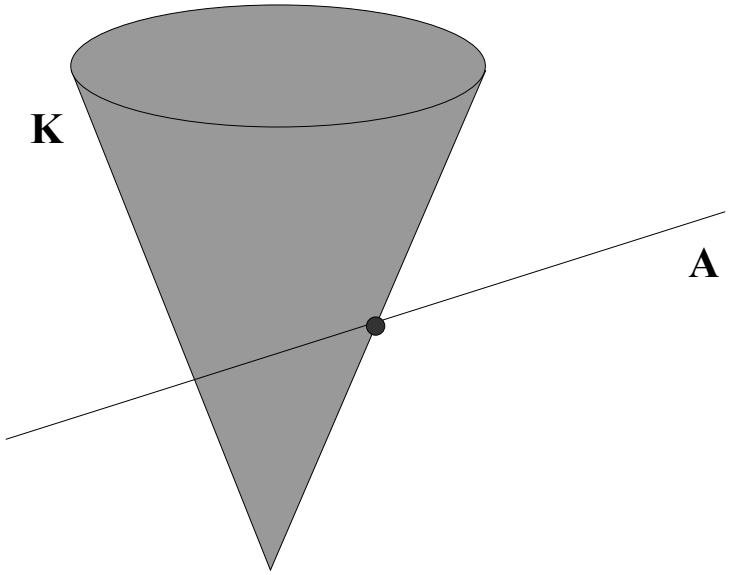

Abbildung 1.1: Menge der Optimallösungen

Voraussetzung 2 impliziert nach Korollar 1 Voraussetzung 1.

Unter Voraussetzung 1 gilt

$$(x, s) \text{ ist Optimallösung von } (P) \text{ und } (D) \Leftrightarrow (x, s) \in \mathbf{A} \cap \mathbf{K}. \tag{1.4}$$

Die Menge der Optimallösungen des primal-dualen Paares von Programmen ist der Schnitt des affinen Raumes $\mathbf{A}$ mit dem konvexen Kegel $\mathbf{K}$.

Sei ein euklidischer Raum $(\bar{E}, \langle ., . \rangle_{\bar{E}})$ gegeben. Wir setzen wieder $\|z\|_2 := \sqrt{\langle z, z \rangle_{\bar{E}}}$ für alle $z \in \bar{E}$.
Für eine nicht-leere abgeschlossene konvexe Menge $C \subseteq \bar{E}$ und $\bar{z} \in \bar{E}$ sei

$$\text{dist}(\bar{z}, C) := \inf_{z \in C} \|z - \bar{z}\|_2$$

die Distanz von $\bar{z}$ zu C. Da $(\bar{E}, \langle ., . \rangle_{\bar{E}})$ ein reeller endlichdimensionaler Hilbert-Raum, C eine abgeschlossene Menge und $\|.\|_2$ eine streng konvexe Funktion ist, existiert ein eindeutig bestimmtes Element $\tilde{z} \in C$ mit $\text{dist}(\bar{z}, C) = \|\tilde{z} - \bar{z}\|_2 = \min_{z \in C} \|z - \bar{z}\|_2$. Die Distanzfunktion $z \mapsto \text{dist}(z, C)$ ist konvex. Wir setzen:

$$\Pi_C(\bar{z}) = \tilde{z} = \text{argmin}_{z \in C} \|z - \bar{z}\|_2.$$

Sind $(E_1, \langle ., . \rangle_{E_1})$ und $(E_2, \langle ., . \rangle_{E_2})$ reelle endlichdimensionale Hilbert-Räume, dann ist auch $(\bar{E}, \langle ., . \rangle_{\bar{E}})$ mit $\bar{E} := E_1 \times E_2$ und dem Skalarprodukt

$$\langle ., . \rangle_{\bar{E}} := \langle ., . \rangle_{E_1} + \langle ., . \rangle_{E_2}$$

ein reeller endlichdimensionaler Hilbert-Raum.

Sind Programme (P) und (D) gegeben, dann betrachten wir für $z \in \bar{E} := E \times E$ versehen mit dem durch $\langle ., . \rangle_E$ induzierten Skalaprodukt $\langle ., . \rangle_{\bar{E}}$ die konvexe Funktion

$$\phi(z) = \frac{1}{2}(\mathrm{dist}(z, \mathbf{A})^2 + \mathrm{dist}(z, \mathbf{K})^2).$$

Nach (1.4) gilt für $z := (x, s) \in E \times E$

$$(x, s) \text{ ist Optimallösung von } (P) \text{ und } (D) \Leftrightarrow \phi(z) = 0.$$

Wir beweisen einen Hilfssatz, welcher in ähnlicher Form bereits in [9] gezeigt wurde:

Lemma 1. *Sei $C \subseteq \bar{E}$ eine nicht-leere konvexe abgeschlossene Menge. Sei weiterhin für $z \in \bar{E}$*

$$\phi_C(z) := \frac{1}{2}\mathrm{dist}(z, C)^2 = \frac{1}{2}\|z - \Pi_C(z)\|_2^2.$$

ϕ_C ist differenzierbar und es gilt

$$\nabla\phi_C(z) = z - \Pi_C(z),$$

wobei wir hier und auch im Folgenden die Abbildung $D\phi_C(z) : \bar{E} \to I\!R$ für ein $z \in \bar{E}$ mit einem entsprechenden (eindeutigen) Element $\nabla\phi_C(z) \in \bar{E}$ identifizieren, so dass $D\phi_C(z)[h] = \langle \nabla\phi_C(z), h \rangle_{\bar{E}}$ für alle $h \in \bar{E}$ gilt.

Beweis. Seien $z, h \in \bar{E}$ gegeben. Wir setzen $\hat{z} := \Pi_C(z)$. Wir zeigen

$$\lim_{h \to 0} \frac{|\phi_C(z + h) - \phi_C(z) - \langle z - \hat{z}, h \rangle|}{\|h\|_2} = 0.$$

Sei $\hat{z}_h := \Pi_C(z + h)$. Es gilt

$$2\phi_C(z + h) = \|z + h - \hat{z}_h\|_2^2 \leq \|z + h - \hat{z}\|_2^2 = \underbrace{\|z - \hat{z}\|_2^2}_{=2\phi_C(z)} + 2\langle z - \hat{z}, h \rangle + \|h\|_2^2.$$

Daraus folgt $\phi_C(z + h) - \phi_C(z) - \langle z - \hat{z}, h \rangle \leq \frac{1}{2}\|h\|_2^2$. Andererseits gilt

$$2\phi_C(z+h) = \|z+h-\hat{z}+\hat{z}-\hat{z}_h\|_2^2 = \|z-\hat{z}\|_2^2+2\langle z-\hat{z}, h+\hat{z}-\hat{z}_h\rangle+\|h+\hat{z}-\hat{z}_h\|_2^2.$$

Da Π_C eine Projektion ist folgt $\langle z - \hat{z}, \hat{z} - \hat{z}_h \rangle \geq 0$ und somit

$$\phi_C(z + h) - \phi_C(z) - \langle z - \hat{z}, h \rangle \geq \frac{1}{2}\|h + \hat{z} - \hat{z}_h\|_2^2 \geq 0.$$

Insgesamt erhalten wir

$$0 \leq \phi_C(z + h) - \phi_C(z) - \langle z - \hat{z}, h \rangle \leq \frac{1}{2}\|h\|_2^2,$$

woraus die Behauptung folgt. □

Aus Lemma 1 folgt die Differenzierbarkeit der Funktion $\phi = \phi_{\mathbf{A}} + \phi_{\mathbf{K}}$.

Wie wir bereits angemerkt haben, ist jedes Minimum der Funktion ϕ Optimallösung des primal-dualen Paares (P) und (D). Da die Funktion konvex und differenzierbar ist, lassen sich eine Reihe von Verfahren verwenden um diese zu minimieren. Während allerdings bei ϕ_C ein voller Schritt in Richtung des negativen Gradienten genügt um die Funktion zu minimieren, kann selbiges Verfahren (steilster Abstieg) zur Minimierung von $\phi = \phi_{\mathbf{A}} + \phi_{\mathbf{K}}$ extrem langsam konvergieren. Aus numerischer Sicht müssen neben der Konvergenzordnung auch der Speicherbedarf und der Rechenaufwand des verwendeten Verfahrens berücksichtigt werden. Wichtige Faktoren sind hierbei die Projektion $\Pi_{\mathbf{A}}$ und vor allem $\Pi_{\mathbf{K}}$. Letztere ist für lineare Programme und für semidefinite Programme bis zu einer gegebenen Rechengenauigkeit bestimmbar.

Wir werden uns im Folgenden zunächst mit der Projektion auf die affine Menge $\mathbf{A}$ befassen: Sei M ein Untervektorraum von E. Da $E = M \oplus M^{\perp}$, gilt $z = \Pi_M(z) + \Pi_{M^{\perp}}(z)$ für alle $z \in E$. Es folgt somit $\mathcal{L} + b = \mathcal{L} + \Pi_{\mathcal{L}}(b) + \Pi_{\mathcal{L}^{\perp}}(b) = \mathcal{L} + \Pi_{\mathcal{L}^{\perp}}(b)$ und $\mathcal{L}^{\perp} + c = \mathcal{L}^{\perp} + \Pi_{\mathcal{L}}(c)$. Wir setzen

$$\mathbf{A_1} := (\mathcal{L} + b) \times (\mathcal{L}^{\perp} + c) = (\mathcal{L} + \Pi_{\mathcal{L}^{\perp}}(b)) \times (\mathcal{L}^{\perp} + \Pi_{\mathcal{L}}(c)).$$

Seien weiterhin

$$\begin{aligned} \mathbf{A_2} &:= \{(x, s) \mid \langle c, x\rangle + \langle b, s\rangle = \langle b, c\rangle\}, \\ \mathbf{L_2} &:= \{(x, s) \mid \langle \Pi_{\mathcal{L}}(c), x\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(b), s\rangle = 0\}. \end{aligned}$$

Für $(\bar{x}, \bar{s}) \in \mathbf{A_1}$ gilt:

$$\exists \hat{x} \in \mathcal{L},\ \hat{s} \in \mathcal{L}^{\perp} : \bar{x} = \hat{x} + \Pi_{\mathcal{L}^{\perp}}(b),\ \bar{s} = \hat{s} + \Pi_{\mathcal{L}}(c).$$

Somit folgt

$$\begin{aligned} (\bar{x}, \bar{s}) \in \mathbf{A_2} \Leftrightarrow\ & \langle c, \bar{x}\rangle + \langle b, \bar{s}\rangle = \langle b, c\rangle \\ \Leftrightarrow\ & \langle \Pi_{\mathcal{L}}(c) + \Pi_{\mathcal{L}^{\perp}}(c), \bar{x}\rangle + \langle \Pi_{\mathcal{L}}(b) + \Pi_{\mathcal{L}^{\perp}}(b), \bar{s}\rangle \\ & = \langle \Pi_{\mathcal{L}}(b) + \Pi_{\mathcal{L}^{\perp}}(b), \Pi_{\mathcal{L}}(c) + \Pi_{\mathcal{L}^{\perp}}(c)\rangle \\ \Leftrightarrow\ & \langle \Pi_{\mathcal{L}}(c), \bar{x}\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(c), \hat{x} + \Pi_{\mathcal{L}^{\perp}}(b)\rangle \\ & + \langle \Pi_{\mathcal{L}}(b), \hat{s} + \Pi_{\mathcal{L}}(c)\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(b), \bar{s}\rangle \\ & = \langle \Pi_{\mathcal{L}}(b), \Pi_{\mathcal{L}}(c)\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(b), \Pi_{\mathcal{L}^{\perp}}(c)\rangle \\ \Leftrightarrow\ & \langle \Pi_{\mathcal{L}}(c), \bar{x}\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(c), \Pi_{\mathcal{L}^{\perp}}(b)\rangle \\ & + \langle \Pi_{\mathcal{L}}(b), \Pi_{\mathcal{L}}(c)\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(b), \bar{s}\rangle \\ & = \langle \Pi_{\mathcal{L}}(b), \Pi_{\mathcal{L}}(c)\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(b), \Pi_{\mathcal{L}^{\perp}}(c)\rangle \\ \Leftrightarrow\ & \langle \Pi_{\mathcal{L}}(c), \bar{x}\rangle + \langle \Pi_{\mathcal{L}^{\perp}}(b), \bar{s}\rangle = 0 \\ \Leftrightarrow\ & (\bar{x}, \bar{s}) \in \mathbf{L_2}. \end{aligned}$$

Wir haben damit $\mathbf{A} = \mathbf{A_1} \cap \mathbf{A_2} = \mathbf{A_1} \cap \mathbf{L_2}$ gezeigt.
Sei $\mathbf{L_1} := \mathcal{L} \times \mathcal{L}^\perp$ und $\mathbf{b_1} := (\Pi_{\mathcal{L}^\perp}(b), \Pi_{\mathcal{L}}(c))$. Dann gilt $\mathbf{A_1} = \mathbf{L_1} + \mathbf{b_1}$. Sei weiterhin $(\tilde{x}, \tilde{s}) \in \mathbf{L_1}^\perp$. Es gilt dann $\langle (\tilde{x}, \tilde{s}), (x, s) \rangle_{\bar{E}} = 0$ für alle $(x, s) \in \mathbf{L_1}$. Damit folgt insbesondere $(\tilde{x}, \tilde{s}) \in \mathbf{L_2}$. Es gilt somit $\mathbf{L_1}^\perp \subseteq \mathbf{L_2}$. Damit haben wir alle Voraussetzungen nachgewiesen, die für die folgenden allgemeinen Aussagen benötigt werden:

Lemma 2. *Sei $(\bar{E}, \langle ., . \rangle_{\bar{E}})$ ein euklidischer Raum, $M \subseteq \bar{E}$ ein Untervektorraum und $m \in \bar{E}$. Für den affinen Raum $N := M + m$ und $z \in \bar{E}$ gilt*

$$\Pi_N(z) = \Pi_M(z - m) + m.$$

Beweis. Es gilt $\Pi_M(z - m) + m \in N$. Da $\Pi_N(z) - m \in M$, ist die Annahme

$$\begin{aligned} & \|\Pi_N(z) - z\|_2 < \|\Pi_M(z - m) + m - z\|_2 \\ \Leftrightarrow \quad & \|\Pi_N(z) - m - (z - m)\|_2 < \|\Pi_M(z - m) - (z - m)\|_2 \end{aligned}$$

offensichtlich falsch. Aus der Eindeutigkeit von $\Pi_N(z)$ folgt die Behauptung. □

Satz 2. *Seien M_1 und M_2 Untervektorräume eines euklidischen Raumes $(\bar{E}, \langle ., . \rangle_{\bar{E}})$ und es gelte $M_1^\perp \subseteq M_2$. Seien weiterhin $m_1, m_2 \in \bar{E}$ gegeben. Für $N_1 := M_1 + m_1$ und $N_2 := M_2 + m_2$ gilt dann $N_1 \cap N_2 \neq \emptyset$ und für alle $z \in \bar{E}$ gilt*

$$\Pi_{N_1 \cap N_2}(z) = \Pi_{N_1}(\Pi_{N_2}(z)) = \Pi_{N_2}(\Pi_{N_1}(z)).$$

Beweis. Es gilt $M_1 \cap M_2 \neq \emptyset$. Aus der für beliebige Untervektorräume U, V gültigen Aussage $U \subseteq V \Leftrightarrow V^\perp \subseteq U^\perp$ folgt $M_1^\perp \subseteq M_2 \Leftrightarrow M_2^\perp \subseteq M_1$. Daraus folgt

$$\begin{array}{lll} \Pi_{M_1}(\Pi_{M_2}(z)) & = \Pi_{M_1}(z - \Pi_{M_2^\perp}(z)) & = \Pi_{M_1}(z) - \Pi_{M_1}(\Pi_{M_2^\perp}(z)) \\ & = \Pi_{M_1}(z) - \Pi_{M_2^\perp}(z) & = (z - \Pi_{M_1^\perp}(z)) - (z - \Pi_{M_2}(z)) \\ & = \Pi_{M_2}(z) - \Pi_{M_1^\perp}(z) & = \Pi_{M_2}(z) - \Pi_{M_2}(\Pi_{M_1^\perp}(z)) \\ & = \Pi_{M_2}(z - \Pi_{M_1^\perp}(z)) & = \Pi_{M_2}(\Pi_{M_1}(z)). \end{array}$$

Dabei haben wir ausgenutzt, dass die Projektion Π_{M_j} auf den Untervektorraum M_j eine lineare Funktion ist und dass $z = \Pi_{M_j}(z) + \Pi_{M_j{}^\perp}(z)$ gilt.
Als Folgerung erhalten wir $\Pi_{M_1}(\Pi_{M_2}(z)) = \Pi_{M_2}(\Pi_{M_1}(z)) \in M_1 \cap M_2$.

Angenommen $\hat{z} := \Pi_{M_1 \cap M_2}(z) \neq \Pi_{M_1}(\Pi_{M_2}(z)) =: \tilde{z}$. Dann gilt

$$\|z - \hat{z}\|_2^2 < \|z - \tilde{z}\|_2^2.$$

Mit

$$\begin{aligned}\|z-\hat{z}\|_2^2 &= \|z-\Pi_{M_2}(z)\|_2^2 + 2\langle \underbrace{z-\Pi_{M_2}(z)}_{=\Pi_{M_2^\perp}(z)}, \underbrace{\Pi_{M_2}(z)-\hat{z}}_{\in M_2}\rangle + \|\Pi_{M_2}(z)-\hat{z}\|_2^2 \\ &= \|z-\Pi_{M_2}(z)\|_2^2 + \|\Pi_{M_2}(z)-\hat{z}\|_2^2, \\ \|z-\tilde{z}\|_2^2 &= \|z-\Pi_{M_2}(z)\|_2^2 + 2\langle \underbrace{z-\Pi_{M_2}(z)}_{=\Pi_{M_2^\perp}(z)}, \underbrace{\Pi_{M_2}(z)-\tilde{z}}_{\in M_2}\rangle + \|\Pi_{M_2}(z)-\tilde{z}\|_2^2 \\ &= \|z-\Pi_{M_2}(z)\|_2^2 + \|\Pi_{M_2}(z)-\tilde{z}\|_2^2\end{aligned}$$

folgt daraus $\|\Pi_{M_2}(z)-\hat{z}\|_2^2 < \|\Pi_{M_2}(z)-\tilde{z}\|_2^2$. Aus der Tatsache, dass $\tilde{z}$ die Projektion von $\Pi_{M_2}(z)$ auf M_1 ist und $\hat{z} \in M_1$ gilt, folgt der gesuchte Widerspruch. Wir haben damit

$$\Pi_{M_1\cap M_2}(z) = \Pi_{M_1}(\Pi_{M_2}(z)) = \Pi_{M_2}(\Pi_{M_1}(z)) \tag{1.5}$$

gezeigt.

Es gilt $\bar{E} = M_1 + M_1^\perp$ und

$$N_1 \cap N_2 = (M_1+m_1)\cap(M_2+m_2) = m_1 + \{M_1 \cap ((m_2-m_1)+M_2)\}.$$

Für $\hat{m} := m_2 - m_1$ folgt

$$\hat{m} = \underbrace{\mu_1}_{\in M_1} + \underbrace{\mu_2}_{\in M_1^\perp \subseteq M_2} \quad \text{und damit } \underbrace{\hat{m}-\mu_2}_{\in \hat{m}+M_2} = \underbrace{\mu_1}_{\in M_1} \in M_1 \cap (\hat{m}+M_2).$$

Also ist $N_1 \cap N_2 \neq \emptyset$.
Somit exisitert ein $m \in N_1 \cap N_2$, so dass $N_1 = M_1 + m$ und $N_2 = M_2 + m$ und damit $N_1 \cap N_2 = M_1 \cap M_2 + m$ gilt. Mit Lemma 2 und (1.5) folgt

$$\begin{aligned}\Pi_{N_1\cap N_2}(z) &= \Pi_{M_1\cap M_2}(z-m)+m \\ &= \Pi_{M_1}(\Pi_{M_2}(z-m))+m \\ &= \Pi_{M_1}((\Pi_{M_2}(z-m)+m)-m)+m \\ &= \Pi_{M_1}(\Pi_{N_2}(z)-m)+m \\ &= \Pi_{N_1}(\Pi_{N_2}(z)).\end{aligned}$$

Dito $\Pi_{N_1\cap N_2}(z) = \Pi_{N_2}(\Pi_{N_1}(z))$. □

Erfüllt ein gegebenes Paar (P) und (D) Voraussetzung 1, dann folgt für $\bar{E} := E \times E$ und $\mathbf{L} := \mathbf{L_1} \cap \mathbf{L_2}$ aus Satz 2:

Es gilt $\mathbf{b} := \mathbf{b_1} = (\Pi_{\mathcal{L}^\perp}(b), \Pi_{\mathcal{L}}(c)) \in \mathbf{A_1} \cap \mathbf{L_2} = \mathbf{A}$ und somit $\mathbf{A} = \mathbf{L} + \mathbf{b}$. Weiterhin lässt sich die Projektion $\Pi_{\mathbf{A}}$ mit den Formeln $\Pi_{\mathbf{A}}(z) = \Pi_{\mathbf{L}}(z-\mathbf{b}) + \mathbf{b}$

und $\Pi_{\mathbf{L}}(z) = \Pi_{\mathbf{L_2}}(\Pi_{\mathbf{L_1}}(z))$ leicht auswerten, falls die Projektion auf $\mathcal{L}$ einfach zu bestimmen ist: Sei

$$\mathcal{L} = \{x \in E \mid \mathcal{A}(x) = 0\}$$

für eine lineare Abbildung $\mathcal{A} : E \to \mathbb{R}^d$, $d \leq \dim(E)$, dann folgt

$$\mathcal{L}^\perp = \{\mathcal{A}^*(y) \mid y \in \mathbb{R}^d\}.$$

Die Projektionen auf $\mathcal{L}$ und $\mathcal{L}^\perp$ lassen sich somit unter der Voraussetzung, dass $\mathcal{A}\mathcal{A}^*$ invertierbar ist, in folgender Form angeben:

$$\begin{aligned} \Pi_{\mathcal{L}}(x) &= x - \mathcal{A}^*((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(x))) \\ \Pi_{\mathcal{L}^\perp}(s) &= \mathcal{A}^*((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(s))). \end{aligned}$$

Für $z := (x, s)$ gilt dann wegen $\mathbf{L_1} = \mathcal{L} \times \mathcal{L}^\perp$

$$\Pi_{\mathbf{L_1}}(z) = (\Pi_{\mathcal{L}}(x), \Pi_{\mathcal{L}^\perp}(s)).$$

Da $\mathbf{L_2}$ eine Hyperebene ist, kann man die Projektion auf diesen Raum ganz einfach berechnen. Für einen beliebigen affinen Raum der Form $\mathcal{H} = \{z \in \bar{E} \mid \langle \alpha, z \rangle_{\bar{E}} = \beta\}$ mit $\alpha \in \bar{E}$, $\beta \in \mathbb{R}$ gilt $\Pi_{\mathcal{H}}(z) = z - \frac{\langle \alpha, z \rangle - \beta}{\langle \alpha, \alpha \rangle} \alpha$. Somit folgt

$$\Pi_{\mathbf{L_2}}(z) = z - \frac{\langle (\Pi_{\mathcal{L}}(c), \Pi_{\mathcal{L}^\perp}(b)), z \rangle}{\|(\Pi_{\mathcal{L}}(c), \Pi_{\mathcal{L}^\perp}(b))\|_2^2} (\Pi_{\mathcal{L}}(c), \Pi_{\mathcal{L}^\perp}(b)).$$

Im Allgemeinen ist die Berechnung der Projektion auf den Schnitt von zwei affinen Räumen aufwändiger.

Die Schwierigkeit der Berechnung von $\Pi_{\mathbf{K}}$ hängt natürlich von den Kegeln $\mathcal{K}$ und $\mathcal{K}^D$ ab. Sind (P) und (D) lineare Programme, dann ist $E = \mathbb{R}^n$ und $\mathcal{K} = \mathcal{K}^D = \mathbb{R}^n_{\geq 0}$. In diesem Fall ist die Projektion wegen $\Pi_{\mathcal{K}}(x) = \max\{0, x\}$ leicht zu berechnen. Auch im Fall der semidefiniten Programme gilt die Projektion auf $\mathbf{K}$ als berechenbar. Sie setzt jedoch eine Eigenwertzerlegung voraus, die nicht als billig bezeichnet werden kann. Wir werden auf diesen Fall später ausführlich eingehen.

1.3 Augmentierte primal-duale Methode

Basierend auf der bisherigen Herleitung geben wir im Folgenden ein Verfahren zur Bestimmung der Optimallösung eines primal-dualen Paares (P) und (D) von konischen Programmen an. Die bisherigen und die folgenden Überlegungen entsprechen im Wesentlichen der Herleitung in [9].

Wir betrachten noch einmal die Funktion $\phi = \phi_{\mathbf{A}} + \phi_{\mathbf{K}}$. Anstatt bei der Minimierung von ϕ den gesamten Raum $\bar{E}$ zu betrachten, kann man sich auf eine

„sinnvolle" Teilmenge beschränken. Entsprechend der Definition von ϕ bietet sich eine Einschränkung auf den Raum $\mathbf{A}$ oder $\mathbf{K}$ an. Methoden, die für spezielle Problemklassen mit Elementen aus dem Kegel $\mathbf{K}$ arbeiten, sind z.B. die Boundary Point Methode (siehe [12]) oder Innere-Punkte-Methoden (siehe z.B. [10]). Wir betrachten dagegen die Einschränkung auf $\mathbf{A}$:

Wir setzen $\tilde{\phi} := \phi_{|\mathbf{A}}$. Es folgt nun entsprechend (1.4) für $z = (x, s) \in \mathbf{A}$:

$$(x, s) \text{ ist Optimallösung von } (P) \text{ und } (D) \Leftrightarrow \tilde{\phi}(z) = 0.$$

Seien euklidische Räume $(U, \langle ., . \rangle_U)$ und $(V, \langle ., . \rangle_V)$ gegeben. Sei weiterhin $N = M + m$, wobei M ein Unterraum von U und $m \in U$ sei. Sei $\alpha : N \to V$ eine Abbildung und $u \in N$ gegeben. Wir sagen, dass α in u differenzierbar ist, falls eine lineare Abbildung $L : M \to V$ existiert, so dass

$$\lim_{\substack{h \to 0 \\ h \in M}} \frac{\|\alpha(u + h) - \alpha(u) - L(h)\|_{2(V)}}{\|h\|_{2(U)}} = 0$$

erfüllt ist. Dies is die naheliegende Erweiterung des Differenzierbarkeitsbegriffes auf affine Räume. Unter Verwendung dieser Erweiterung ist für uns das folgende Lemma von Bedeutung, welches in ähnlicher Weise bereits in [9] gezeigt wurde:

Lemma 3. *Sei $(\bar{E}, \langle ., . \rangle_{\bar{E}})$ ein euklidischer Raum, $C \subseteq \bar{E}$ eine nicht-leere konvexe abgeschlossene Menge, M ein Unterraum von $\bar{E}$ und $m \in \bar{E}$. Sei weiterhin $N := M + m$. Für $z \in N$ sei*

$$\tilde{\phi}_C(z) := \frac{1}{2}\text{dist}(z, C)^2 = \frac{1}{2}\|z - \Pi_C(z)\|_2^2.$$

Die Funktion $\tilde{\phi}_C$ ist auf N differenzierbar und es gilt

$$\nabla\tilde{\phi}_C(z) = z - \Pi_N(\Pi_C(z)) = \Pi_M(z - \Pi_C(z)),$$

falls wir die Abbildung $D\tilde{\phi}_C(z) : M \to I\!R$ an der Stelle $z \in N$ mit einem entsprechenden (eindeutigen) Element $\nabla\tilde{\phi}_C(z) \in M$ identifizieren, so dass $D\tilde{\phi}_C(z)[h] = \langle \nabla\tilde{\phi}_C(z), h \rangle_{\bar{E}}$ für alle $h \in M$ gilt.

Beweis. Es gilt nach Lemma 2 für $z \in N$

$$\begin{aligned} z - \Pi_N(\Pi_C(z)) &= \underbrace{z - m}_{\in M} + m - (\Pi_M(\Pi_C(z) - m) + m) \\ &= \Pi_M(z - m - (\Pi_C(z) - m)) = \Pi_M(z - \Pi_C(z)). \end{aligned}$$

Seien $z \in N$ und $h \in M$ gegeben. Wir setzen $\hat{z} := \Pi_C(z)$ und zeigen analog zu Lemma 1

$$\lim_{h \to 0} \frac{|\tilde{\phi}_C(z + h) - \tilde{\phi}_C(z) - \langle \Pi_M(z - \hat{z}), h \rangle|}{\|h\|_2} = 0.$$

Sei $\hat{z}_h := \Pi_C(z+h)$. Es gilt wegen $h = \Pi_M(h)$ und $\langle u, \Pi_M(v)\rangle = \langle \Pi_M(u), v\rangle$ für alle $u, v \in \bar{E}$

$$2\tilde{\phi}_C(z+h) = \|z+h-\hat{z}_h\|_2^2 \leq \|z+h-\hat{z}\|_2^2 = \underbrace{\|z-\hat{z}\|_2^2}_{=2\tilde{\phi}_C(z)} + 2\ \underbrace{\langle z-\hat{z}, h\rangle}_{=\langle \Pi_M(z-\hat{z}),h\rangle} + \|h\|_2^2.$$

Daraus folgt $\tilde{\phi}_C(z+h) - \tilde{\phi}_C(z) - \langle \Pi_M(z-\hat{z}), h\rangle \leq \frac{1}{2}\|h\|_2^2$. Andererseits gilt

$$2\tilde{\phi}_C(z+h) = \|z+h-\hat{z}+\hat{z}-\hat{z}_h\|_2^2 = \|z-\hat{z}\|_2^2 + 2\ \underbrace{\langle z-\hat{z}, h+\hat{z}-\hat{z}_h\rangle}_{=\langle \Pi_M(z-\hat{z}),h\rangle + \langle z-\hat{z}, \hat{z}-\hat{z}_h\rangle} + \|h+\hat{z}-\hat{z}_h\|_2^2.$$

Wegen $\langle z-\hat{z}, \hat{z}-\hat{z}_h\rangle \geq 0$ folgt

$$\tilde{\phi}_C(z+h) - \tilde{\phi}_C(z) - \langle \Pi_M(z-\hat{z}), h\rangle \geq \frac{1}{2}\|h+\hat{z}-\hat{z}_h\|_2^2 \geq 0.$$

Wir erhalten

$$0 \leq \tilde{\phi}_C(z+h) - \tilde{\phi}_C(z) - \langle \Pi_M(z-\hat{z}), h\rangle \leq \frac{1}{2}\|h\|_2^2.$$

Daraus folgt die Behauptung. □

Für unsere bezüglich (P) und (D) definierte Funktion $\tilde{\phi} = \phi_{|\mathbf{A}}$ folgt aus Lemma 3:

$$\nabla\tilde{\phi}(z) = z - \Pi_{\mathbf{A}}(\Pi_{\mathbf{K}}(z)) = \Pi_{\mathbf{L}}(z - \Pi_{\mathbf{K}}(z)).$$

Durch die Einschränkung auf den Raum $\mathbf{A}$ wird die Auswertung des Gradienten nicht teurer. Außerdem lassen sich alle Verfahren wie z.B. das Newton-, cg- oder das BFGS-Verfahren (bei geeigneter Wahl der Startmatrix) zur Minimierung von $\tilde{\phi}$ nutzen, sofern man sie auf dem Raum $\bar{E} = E \times E$ entsprechend der Definition für den Raum $\mathbb{R}^n$ verwendet.

Wir geben nun einen allgemeinen Algorithmus an, mit dem für ein gegebenes primal duales Paar (P) und (D), welches Voraussetzung 1 erfüllt, die Optimallösung approximiert wird.

Algorithmus 1. *(Minimierungsverfahren mit linesearch)*
Gegeben sei ein $z^0 \in \mathbf{A}$. Setze $\Delta z^0 := -\nabla\tilde{\phi}(z^0)$ und $k := 0$.

1. *Bestimme $\lambda_k := \operatorname{argmin}_{\lambda>0}\tilde{\phi}(z^k + \lambda\Delta z^k)$.*
2. *Setze $z^{k+1} := z^k + \lambda_k\Delta z^k$ und berechne $\nabla\tilde{\phi}(z^{k+1})$.*
3. *Falls $\|\nabla\tilde{\phi}(z^{k+1})\|_2 = 0$, STOPP!*
4. *Bestimme Δz^{k+1} mit Hilfe eines geeigneten Verfahrens wie z.B. cg oder BFGS unter Verwendung von z^{k+1} und $\nabla\tilde{\phi}(z^{k+1})$.*
5. *Setze $k := k+1$ und gehe zu Schritt 1.*

In Schritt 1 wird dabei das globale Minimum von $\tilde{\phi}$ ausgehend von z^k entlang des Pfades Δz^k bestimmt. Diese Prozedur bezeichnet man als (exakte) linesearch.

Kapitel 2

Semidefinite Programme

Die bekannteste Art von konischen Programmen sind die linearen Programme. In diesem Fall ist $E = \mathbb{R}^n$ versehen mit dem Standardskalarprodukt $\langle x, y\rangle = x^T y = \sum_{i=1}^n x_i y_i$. Weiterhin ist $\mathcal{K} = \mathbb{R}^n_{\geq 0}$. Da dieser Kegel selbstdual ist, kann die Projektion auf $\mathbf{K} = \mathcal{K} \times \mathcal{K}^D = \mathcal{K} \times \mathcal{K}$ wegen $\Pi_{\mathcal{K}}(x) = \max\{0, x\}$ in $\mathcal{O}(n)$ Schritten berechnet werden. Ist $\mathcal{L} = \{x \in \mathbb{R}^n \mid Ax = 0\}$, mit $A \in \mathbb{R}^{d\times n}$, $d \leq n$ und $\operatorname{rang}(A) = d$, dann gilt

$$\Pi_{\mathcal{L}}(x) = (I - A^T(AA^T)^{-1}A)x \text{ und } \Pi_{\mathcal{L}^\perp}(s) = A^T(AA^T)^{-1}As.$$

Der wesentliche Aufwand bei der Berechnung der Projektion liegt in der Bestimmung einer Lösung des linearen Gleichungssystems $AA^T y = r$. Dabei wird die Inverse der Matrix AA^T i.Allg. nicht berechnet. Man bestimmt stattdessen z.B. einmalig die Cholesky-Zerlegung $LL^T = AA^T$, wobei der Aufwand dafür in $\mathcal{O}(d^3)$ liegt, und wertet bei der Berechnungung der Projektion ausschließlich Matrix-Vektor-Produkte aus oder löst ein lineares Dreieckssystem. Da $d \leq n$ gilt, lassen sich diese Berechnungen bei gegebener Cholesky-Zerlegung mit $\mathcal{O}(dn)$ Schritten ausführen. Für lineare Programme (P) und (D) ist diese Projektion der teuerste Teil des Verfahrens, welches im vorherigen Kapitel vorgestellt wurde.

Die Kostenverteilung auf die Bestandteile des Verfahrens sieht bei semidefiniten Programmen etwas anders aus. Dies ist einer der Punkte, mit denen wir uns im Folgenden für diese spezielle Klasse von konischen Programmen auseinandersetzen werden. Neben einer genaueren Untersuchung der zu minimierenden Funktion $\tilde{\phi}$ und deren Ableitungen wird das Konvergenzverhalten des Verfahrens sowie dessen Einsatzmöglichkeit für hochdimensionale Programme betrachtet.

2.1 Problemformulierung

Sei

$$\mathcal{S}^n := \{M \in \mathbb{R}^{n\times n} \mid M = M^T\} \text{ und } \mathcal{S}^n_+ := \{M \in \mathcal{S}^n \mid M \succeq 0\}.$$

Es gilt dabei $M \succeq 0 \Leftrightarrow x^T M x \geq 0 \; \forall x \in \mathbb{R}^n$. Mit

$$\langle A, B \rangle_{\mathcal{S}^n} := A \bullet B := \text{spur}(A^T B) = \text{spur}(AB)$$

wird ein Skalarprodukt auf $\mathcal{S}^n$ definiert. Weiterhin ist die Frobeniusnorm einer Matrix A definiert durch $\|A\|_F := \sqrt{\langle A, A \rangle_{\mathcal{S}^n}} = \sqrt{\sum_{i,j=1}^n A_{ij}^2}$. (Der $\bullet$-Operator wird allgemeiner durch $A \bullet B := \text{spur}(A^T B)$ für beliebige Matrizen A, $B \in \mathbb{R}^{m \times n}$ definiert.)
Sei ab jetzt $E := \mathcal{S}^n$. Dann ist $(E, \langle ., . \rangle_E)$ ein eulklidischer Raum mit der zum Skalaprodukt gehörenden Norm $\|.\|_F$. (Abweichend von Kapitel 1 bezeichnen wir diese Norm nicht mit $\|.\|_2$ um Verwechslungen mit der lub_2-Norm zu vermeiden.) Wir setzen weiterhin $\mathcal{K} := \mathcal{S}_+^n$. Man prüft leicht nach, dass $\mathcal{S}_+^n$ ein nicht-leerer konvexer abgeschlossener Kegel ist. Die zu E und $\mathcal{K}$ gehörenden konischen Programme werden als semidefinite Programme bezeichnet und haben die folgende Form:

$$\begin{array}{ll} (P) & \min\{C \bullet X \mid X \in (\mathcal{L} + B) \cap \mathcal{K}\}, \\ (D) & \min\{B \bullet S \mid S \in (\mathcal{L}^\perp + C) \cap \mathcal{K}^D\}. \end{array}$$

Hierbei sind $B, C \in \mathcal{S}^n$ und $\mathcal{L} \subseteq \mathcal{S}^n$ ein Untervektorraum. Wie bei linearen Programmen ist auch in diesem Fall der Kegel selbstdual, d.h. $\mathcal{K}^D = \mathcal{K} = \mathcal{S}_+^n$. Sei $\mathcal{L} = \{X \in \mathcal{S}^n \mid \mathcal{A}(X) = 0\}$ für einen linearen Operator $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^d$, dann erhalten wir die alternative Formulierung

$$\begin{array}{ll} (\bar{P}) & \min\{C \bullet X \mid \mathcal{A}(X) = \bar{b}, X \succeq 0\}, \\ (\bar{D}) & \max\{\bar{b}^T y \mid \mathcal{A}^*(y) + S = C, S \succeq 0\}, \end{array}$$

wobei $\bar{b} = \mathcal{A}(B)$ gelte und $\mathcal{A}^*$ die Bedingung $\mathcal{A}(X)^T y = X \bullet \mathcal{A}^*(y) \; \forall y \in \mathbb{R}^d, X \in \mathcal{S}^n$ erfülle. Wird der Operator $\mathcal{A}$ durch Matrizen $A^{(1)}, \ldots, A^{(d)} \in \mathcal{S}^n$ in der Form

$$\mathcal{A}(X) = \begin{pmatrix} A^{(1)} \bullet X \\ \vdots \\ A^{(d)} \bullet X \end{pmatrix}$$

angegeben, dann erhalten wir aus $\mathcal{A}(X)^T y = \sum_{i=1}^d (A^{(i)} \bullet X) y_i = (\sum_{i=1}^d y_i A^{(i)}) \bullet X = \mathcal{A}^*(y) \bullet X$ gerade

$$\mathcal{A}^*(y) = \sum_{i=1}^d y_i A^{(i)}.$$

Sei $\mathcal{S}_{++}^n := \{M \in \mathcal{S}^n \mid M \succ 0\} \subset \mathcal{S}_+^n$. (Dabei gilt $M \succ 0 \Leftrightarrow x^T M x > 0 \; \forall x \in \mathbb{R}^n, x \neq 0$.)

Voraussetzung 2 läßt sich in diesem Fall folgendermaßen formulieren:
Die Programme (P) und (D) erfüllen die Slater-Bedingung, d.h. es existieren $X \in (\mathcal{L} + B) \cap \mathcal{S}_{++}^n$, $S \in (\mathcal{L}^\perp + C) \cap \mathcal{S}_{++}^n$.

Entsprechend der Festlegung in Abschnitt 1.2 setzen wir

$$\begin{aligned}&\bar{E} := E \times E = \mathcal{S}^n \times \mathcal{S}^n,\\ &\langle (A_1, A_2), (B_1, B_2)\rangle_{\bar{E}} := A_1 \bullet B_1 + A_2 \bullet B_2,\\ &\|(A_1, A_2)\|_{\bar{F}} := \sqrt{\|A_1\|_F^2 + \|A_2\|_F^2}.\end{aligned}$$

Die Mengen $\mathbf{L}$, $\mathbf{A}$ und $\mathbf{K}$ werden ebenfalls wie in Abschnitt 1.2 definiert.

Wir nehmen in allen folgenden Überlegungen an, dass ein primal-duales Paar (P) und (D) gegeben ist und dass die zugehörige Menge $\mathbf{A}$ nicht leer ist. (Voraussetzung 1 impliziert zwar $\mathbf{A} \neq \emptyset$, die Umkehrung gilt aber nicht.)

Die Funktion $\tilde{\phi}$ hat dann für $Z = (X, S) \in \mathbf{A}$ die folgende Gestalt:

$$\tilde{\phi}(Z) = \frac{1}{2}\|(X, S) - \Pi_{\mathbf{K}}((X, S))\|_{\bar{F}}^2 = \frac{1}{2}(\|X - \Pi_{\mathcal{S}_+^n}(X)\|_F^2 + \|S - \Pi_{\mathcal{S}_+^n}(S)\|_F^2).$$

Wie im letzten Kapitel gezeigt wurde, ist die Funktion $\tilde{\phi}$ auf $\mathbf{A}$ konvex und differenzierbar. Wenn wir die Ableitung von $\tilde{\phi}(Z)$ mit dem entsprechenden Element $\nabla\tilde{\phi}(Z) \in \mathbf{L}$ identifizieren (siehe Lemma 3), dann erhalten wir

$$\nabla\tilde{\phi}(Z) = Z - \Pi_{\mathbf{A}}(\Pi_{\mathbf{K}}(Z)) = \Pi_{\mathbf{L}}(Z - \Pi_{\mathbf{K}}(Z)) = \Pi_{\mathbf{L}}\left(\begin{bmatrix} X - \Pi_{\mathcal{S}_+^n}(X) \\ S - \Pi_{\mathcal{S}_+^n}(S)\end{bmatrix}\right).$$

Die in Abschnitt 1.3 vorgestellte Augmentierte primal-duale Methode (im Folgenden: APD-Methode) kann zur Minimierung von $\tilde{\phi}$ verwendet werden um somit eine Optimallösung des primal-dualen Paares (P) und (D) zu approximieren. Um das Verhalten der zu minimierenden Funktion im Verfahren besser zu verstehen, wollen wir im Folgenden die erste und zweite Ableitung von $\tilde{\phi}$ untersuchen. Können wir eine gewisse Glattheit der Funktion nachweisen, dann kann das APD-Verfahren mit entsprechenden Informationen zweiter Ordnung erweitert werden um das Konvergenzverhalten zu verbessern. Insbesondere soll die Anwendbarkeit des Newton-Verfahrens zur Suchschritt- und Schrittweitenbestimmung innerhalb der APD-Methode geprüft werden. Dazu betrachten wir als Nächstes den Gradienten von $\tilde{\phi}$ und insbesondere $\Pi_{\mathbf{K}}$ etwas genauer.

2.2 Die Projektion auf $\mathcal{S}_+^n$

Die Beschaffenheit von $\nabla\tilde{\phi}$ hängt im Wesentlichen von der Funktion $\Pi_{\mathcal{S}_+^n}$ ab. Für eine beliebige Matrix $X \in \mathcal{S}^n$ existiert eine orthogonale Matrix U und eine Diagonalmatrix D, so dass $X = UDU^T$ gilt und $D_{11} \leq \cdots \leq D_{nn}$ die aufsteigend sortierten Eigenwerte von X sind. Gilt $X \in \mathcal{S}_+^n$, dann folgt $\Pi_{\mathcal{S}_+^n}(X) = X$. Gilt $X \notin \mathcal{S}_+^n$, dann sei $k \in \{1, \ldots, n\}$ so gegeben, dass D_{kk} der größte negative Eigenwert von X ist. Aus $UU^T = U^TU = I$ und

$$\|UMU^T\|_F^2 = UMU^T \bullet UMU^T = \text{spur}(UMU^TUMU^T) = \text{spur}(MM) = \|M\|_F^2$$

für ein $M \in \mathcal{S}^n$ lässt sich wegen Invarianz der Eigenwerte unter Ähnlichkeitstransformationen und somit $(U^T \Pi_{\mathcal{S}^n_+}(X) U)_{ii} \geq 0$ für $1 \leq i \leq n$

$$\|X - \Pi_{\mathcal{S}^n_+}(X)\|_F^2 = \|D - U^T \Pi_{\mathcal{S}^n_+}(X) U\|_F^2 \geq \sum_{i=1}^{k} (\underbrace{D_{ii}}_{<0} - \underbrace{(U^T \Pi_{\mathcal{S}^n_+}(X) U)_{ii}}_{\geq 0})^2 \geq \sum_{i=1}^{k} D_{ii}^2$$

folgern. Für $\hat{X} := U \max(0, D) U^T \in \mathcal{S}^n_+$ gilt

$$\|X - \hat{X}\|_F^2 = \sum_{i=1}^{n} (D_{ii} - \max(0, D_{ii}))^2 = \sum_{i=1}^{k} D_{ii}^2.$$

Aus der Definition und Eindeutigkeit der Projektion folgt daher $\hat{X} = \Pi_{\mathcal{S}^n_+}(X)$. Für eine Implementierung der Projektion wird i.Allg. diese Darstellung verwendet. Zunächst muss also die Eigenwertzerlegung U, D von X berechnet werden. Dies ist bei gegebener Rechengenauigkeit in $\mathcal{O}(n^3)$ Schritten möglich (siehe [5]). Anschließend wird die Diagonalmatrix D in $\hat{D} := \max(0, D)$ umgewandelt (Aufwand: $\mathcal{O}(n)$) und schließlich $U \hat{D} U^T$ berechnet (Aufwand: $\mathcal{O}(n^3)$). Der Gesamtaufwand liegt damit in $\mathcal{O}(n^3)$.

Wir geben im Folgenden einige Eigenschaften dieser Projektion an. Wir benötigen zunächst die folgenden Definitionen:

Seien im Folgenden U, V euklidische Räume und sei $\eta : U \to V$ lokal Lipschitzstetig. Aus Rademachers Theorem folgt, dass η fast überall differenzierbar ist, siehe [21].

Definition 4. *Sei*

$$\partial\eta(u) := conv\{\lim_{k\to\infty} D\eta(u^k) \quad | \quad u^k \to u,\ u^k \in \mathrm{Diff}(\eta)\},$$

wobei $\mathrm{Diff}(\eta) = \{u \in U \mid D\eta(u) \text{ existiert}\}$. *$\partial\eta(u)$ heißt verallgemeinerte Jacobi-Matrix von η in u. (Der Begriff Matrix ist wählbar, da ein euklidischer Raum U bzw. V mit* $\dim(U) = n$, $\dim(V) = m$, *isomorph zum $\mathbb{R}^n$ bzw. $\mathbb{R}^m$ ist. Mit dem entsprechenden Isomorphismus lässt sich jedes Element $J \in \partial\eta(u)$, $u \in U$, mit einer Matrix $M_J \in \mathbb{R}^{m \times n}$ identifizieren.)*
Falls η die Ableitung einer reellwertigen Funktion ξ ist, dann heißt $\partial^2\xi(u) := \partial\eta(u)$ die verallgemeinerte Hessematrix von ξ in u.
Weiterhin schreiben wir $\partial^2\xi(u) \succeq 0$ ($\succ 0$), falls die Menge $\partial^2\xi(u)$ ausschließlich symmetrische positiv semidefinite (positiv definite) Elemente enthält. Die Elemente können mit Matrizen in $\mathcal{S}^n$ oder allgemeiner mit symmetrischen Bilinearformen identifiziert werden.

Lemma 4. *Ist $W \subseteq U$ offen und η auf W stetig differenzierbar, dann gilt für alle $u \in W$:*

$$\partial\eta(u) = \{D\eta(u)\}.$$

Beweis. Sei $u \in W$ gegeben. Da η an der Stelle u stetig differenzierbar ist, ist die Funktion insbesondre lokal Lipschitz-stetig. Für alle Folgen $(w^k)_{k\in\mathbb{N}} \subset W$ mit $w^k \to u$ gilt weiterhin $\lim_{k\to\infty} D\eta(w^k) = D\eta(u)$. Für jede Folge $u^k \to u$ gilt schließlich $u^k \in W$ und somit $D\eta(u^k) \to D\eta(u)$. □

Definition 5. *Die Funktion η heißt semiglatt von der Ordnung p $(0 < p \leq 1)$ in $u \in U$, falls:*
i) Es existieren alle Richtungsableitungen von η in u,
ii) Für alle $\tilde{u} \to u$ und $M_{\tilde{u}} \in \partial\eta(\tilde{u})$ gilt

$$\eta(\tilde{u}) - \eta(u) - M_{\tilde{u}}(\tilde{u} - u) = \mathcal{O}(\|\tilde{u} - u\|^{p+1}).$$

Die Funktion η heißt stark semiglatt in $u \in U$, falls Bedingung i) und ii) für $p = 1$ erfüllt ist.
Ist nur i) erfüllt und gilt zusätzlich
ii') Für alle $\tilde{u} \to u$ und $M_{\tilde{u}} \in \partial\eta(\tilde{u})$ gilt

$$\eta(\tilde{u}) - \eta(u) - M_{\tilde{u}}(\tilde{u} - u) = o(\|\tilde{u} - u\|),$$

dann heißt η semiglatt.

Jede Funktion, die semiglatt von der Ordnung $p \in (0, 1]$ ist, ist insbesondere semiglatt.
Aus Definition 5 lassen sich die folgenden Hilfssätze direkt ablesen:

Lemma 5. *Sei $\gamma : U \to V$ eine affine Abbildung, d.h. $\gamma(u) = \alpha(u) + \beta$ für eine lineare Abbildung α und $\beta \in V$, dann ist γ auf U stark semiglatt.*

Beweis. Eine affine Abbildung ist auf U stetig differenzierbar. Damit existieren insbesondere alle Richtungsableitungen für alle $u \in U$. Weiterhin gilt dann $\partial\gamma(\tilde{u}) = \{D\gamma(\tilde{u})\} = \{D\gamma\} = \{\alpha\}$ für alle $\tilde{u} \in U$. Somit folgt für $\tilde{u} \to u$

$$\gamma(\tilde{u}) - \gamma(u) - D\gamma(\tilde{u} - u) = \alpha(\tilde{u}) + \beta - (\alpha(u) + \beta) - \alpha(\tilde{u} - u) = 0.$$

□

Lemma 6. *Seien $\eta : U \to V$ semiglatt von der Ordnung p in $u \in U$ und $\gamma : U \to V$ eine affine Abbildung, $\gamma(u) = \alpha(u) + \beta$. Dann ist auch $\eta + \gamma$ semiglatt von der Ordnung p in $u \in U$.*

Beweis. Es gilt $\partial(\eta + \gamma)(u) = \partial\eta(u) + \alpha$. Mit Lemma 5 schließt man somit, dass $\eta + \gamma$ die in Definition 5 geforderten Eigenschaften *i)* und *ii)* für p besitzt. □

In [25] wurde weiterhin gezeigt:

Lemma 7. *Seien $\bar{E} := \mathcal{S}^{n_1} \times \cdots \times \mathcal{S}^{n_m}$, $U := \mathcal{S}^{d_1} \times \cdots \times \mathcal{S}^{d_k}$ und $V := \mathcal{S}^{t_1} \times \cdots \times \mathcal{S}^{t_l}$. Sei weiterhin $p \in (0, 1]$. Ist $F : \bar{E} \to U$ semiglatt von der Ordnung p in $X \in \bar{E}$ und $G : U \to V$ semiglatt von der Ordnung p in $F(X) \in U$, dann ist $F \circ G$ semiglatt von der Ordnung p in X.*

Satz 3. *Sei* $\bar{E} := \mathcal{S}^{n_1} \times \cdots \times \mathcal{S}^{n_m}$ *für ein* $m \in \mathbb{N}$. *Die Funktion* $\Pi_{\mathcal{S}_+^{n_1} \times \cdots \times \mathcal{S}_+^{n_m}}$ *ist lokal Lipschitz-stetig, stark semiglatt und fast überall differenzierbar.*

Wir gehen an dieser Stelle kurz auf Funktionen ein, welche nur auf einem affinen (oder linearen Teilraum) eines euklidischen Raumes definiert sind: Für Funktionen, die bzgl. der in Abschnitt 1.3 angegebenen Definition differenzierbar sind, gelten alle für diese Arbeit relevanten Differenzierbarkeitsussagen, welche auch für Funktionen gelten, die auf dem gesamten euklidischen Raum definiert und differenzierbar sind. Daher werden wir im Folgenden nicht zwischen der Differenzierbarkeit auf einem affinen Teilraum und dem gesamten Raum unterscheiden, sofern aus dem Kontext hervorgeht, welcher Fall gegeben ist.
Die (lokale) Lipschitz-Stetigkeit wird sinngemäß auf affine Teilräume eingeschränkt, so dass nicht mehr alle Richtungen des zu Grunde liegenden euklidischen Raumes betrachtet werden müssen. Genauso verhält es sich bei der Semiglattheit einer Funktion.
Wir sagen, dass eine symmetrische Bilinearform $B : M \times M \to I\!R$ (wobei M ein linearer Teilraum eines euklidischen Raumes U ist) positiv definit bzw. positv semidefinit ist, falls $B[m,m] > 0 \ \forall m \in M \setminus \{0\}$ bzw. $B[m,m] \geq 0 \ \forall m \in M$ gilt. Eine Teilmenge eines affinen Teilraumes $N \subseteq U$ der Dimension q wird als Nullmenge bezeichnet, falls für eine injektive lineare Abbildung $Q : I\!R^q \to U$ mit $Bild(Q) = N$ die Menge $Q^{-1}(N)$ eine Nullmenge bzgl. des Lebesgue-Maßes ist.

Aus Satz 3 und Lemma 6 folgt, dass die Funktion

$$\begin{array}{rccc} \chi: & \mathcal{S}^n \times \mathcal{S}^n & \to & \mathcal{S}^n \times \mathcal{S}^n, \\ & \begin{pmatrix} X \\ S \end{pmatrix} & \mapsto & \begin{pmatrix} X - \Pi_{\mathcal{S}_+^n}(X) \\ S - \Pi_{\mathcal{S}_+^n}(S) \end{pmatrix}, \end{array}$$

stark semiglatt ist. Nach Lemma 7 ist somit auch die Abbildung

$$\nabla\tilde{\phi}(X,S) = \Pi_{\mathbf{L}}(\chi(X,S))$$

stark semiglatt. Weiterhin übertragen sich sowohl die lokale Lipschitz-Stetigkeit als auch die Differenzierbarkeit. Wir erhalten damit insgesamt:

Korollar 2. *Die Abbildung* $\nabla\tilde{\phi}$ *ist stark semiglatt, lokal Lipschitz-stetig und fast überall differenzierbar auf* $\mathbf{A}$.

Aus [25] kann gefolgert werden, dass die Abbildung χ genau auf der Menge $\mathcal{CS}_\chi := \{(X,S) \in \mathcal{S}^n \times \mathcal{S}^n \mid \det(XS) = 0\}$ nicht differenzierbar ist und somit $\nabla\tilde{\phi}$ höchstens auf der Menge

$$\mathcal{CS}_{\tilde{\phi}} := \{(X,S) \in \mathcal{S}^n \times \mathcal{S}^n \mid \det(XS) = 0\} \cap \mathbf{A}$$

nicht differenzierbar ist. Jede Optimallösung (X^{opt}, S^{opt}) eines primal-dualen Paares (P) und (D) mit dem dazugehörigen y^{opt} (siehe Abschnitt 1.1) befindet sich

wegen

$$\begin{aligned}
& C \bullet X^{opt} + B \bullet S^{opt} - B \bullet C = 0 \\
\Leftrightarrow \quad & C \bullet X^{opt} + B \bullet (C - \mathcal{A}^*(y^{opt})) - B \bullet C = 0 \\
\Leftrightarrow \quad & C \bullet X^{opt} - B \bullet \mathcal{A}^*(y^{opt}) = 0 \\
\Leftrightarrow \quad & C \bullet X^{opt} - \bar{b}^T y^{opt} = 0 \\
\Leftrightarrow \quad & C \bullet X^{opt} - X^{opt} \bullet \mathcal{A}^*(y^{opt}) = 0 \\
\Leftrightarrow \quad & X^{opt} \bullet S^{opt} = 0 \\
\Leftrightarrow^{1} \quad & X^{opt} S^{opt} = 0.
\end{aligned} \tag{2.1}$$

grundsätzlich in der kritischen Menge $\mathcal{CS}_{\tilde{\phi}}$. Die Verwendung des normalen Newton-Verfahrens ist daher problematisch. Wir haben in diesem Abschnitt jedoch gezeigt, dass die Ableitung von $\tilde{\phi}$ gewisse Glattheitseigenschaften besitzt. Diese werden uns bei der weiteren Betrachtung der APD-Methode für den semidefiniten Fall nützlich sein.

2.3 Die verallgemeinerte Hessematrix von $\tilde{\phi}$

Unser Ziel ist es, die Funktion $\tilde{\phi}$ mit einem geeigneten Verfahren zu minimieren. Eine Möglichkeit dazu ist Algorithmus 1. Dort wurde allerdings das Verfahren zur Bestimmung der Suchrichtung offen gelassen. Das Verfahren des steilsten Abstiegs und das cg-Verfahren, welche ausschließlich die erste Ableitung von $\tilde{\phi}$ nutzen, sind zu diesem Zweck wählbar – die Konvergenzrate dieser Verfahren kann aber sehr schlecht sein. Das Newton-Verfahren, auf das wir später noch ausführlicher eingehen werden, verwendet zusätzlich die zweite Ableitung der zu minimierenden Funktion und ist unter bestimmten Voraussetzungen lokal quadratisch konvergent. Die Voraussetzungen beinhalten allerdings, dass die Funktion im Minimum mindestens zweimal stetig differenzierbar ist. Am Ende von Abschnitt 2.2 haben wir uns aber überlegt, dass dies i.Allg. für ein Minimum von $\tilde{\phi}$ nicht gewährleistet werden kann. In [20] wurde für solche Fälle das folgende verallgemeinerte Newton-Verfahren angegeben:

Sei U ein euklidischer Raum und $\eta : U \to U$ lokal Lipschitz-stetig. Gesucht ist ein $u \in U$ mit $\eta(u) = 0$.

Algorithmus 2 (Verallgemeinertes Newton-Verfahren)**.** *Sei $u^0 \in U$ gegeben. Für $k = 0, 1, 2, \ldots$*

1. *Wähle $M_k \in \partial\eta(u^k)$.*
2. *Setze $u^{k+1} := u^k - M_k^{-1}\eta(u^k)$.*

[1] Wegen $X^{opt} \succeq 0$ und $S^{opt} \succeq 0$ existieren Matrizen $\hat{X} \succeq 0$ und $\hat{S} \succeq 0$ mit $\hat{X}\hat{X} = X^{opt}$ und $\hat{S}\hat{S} = S^{opt}$. Es folgt somit $0 = \text{spur}(X^{opt}S^{opt}) = \text{spur}(\hat{S}\hat{X}\hat{X}\hat{S}) = \|\hat{X}\hat{S}\|_F^2$ und damit $\hat{X}\hat{S} = 0$. Wir erhalten $X^{opt}S^{opt} = \hat{X}(\hat{X}\hat{S})\hat{S} = 0$. Die andere Implikation folgt direkt aus der Definition von $\bullet$.

Ohne weitere Voraussetzungen ist die Wohldefiniertheit geschweige denn die Konvergenz des obigen Verfahrens nicht gegeben. In [20] wird jedoch der folgende Satz gezeigt:

Satz 4. *Sei u^* mit $\eta(u^*) = 0$ gegeben. Sei η lokal Lipschitz-stetig und semiglatt von der Ordnung p in u^* und es seien alle $M \in \partial\eta(u^*)$ invertierbar. Dann existiert eine Umgebung $N(u^*)$ von u^*, so dass Algorithmus 2 für alle $u^0 \in N(u^*)$ wohldefiniert ist und $u^k \to u^*$ gilt. Die Konvergenzrate ist dabei von der Ordnung $p+1$.*

Um zu prüfen ob bei der Minimierung von $\tilde{\phi}$ auf $\mathbf{A}$ die Voraussetzungen von Satz 4 erfüllt sind, müssen wir uns zunächst mit der verallgemeinerten Hessematrix von $\tilde{\phi}$ an einer Minimalstelle Z^{opt} befassen. Da es sich bei $\partial^2\tilde{\phi}(Z^{opt})$ um die konvexe Hülle der Grenzwerte von zweiten Ableitungen von $\tilde{\phi}$ handelt, betrachten wir zunächst den Bereich in $\mathcal{S}^n \times \mathcal{S}^n \cap \mathbf{A}$, auf dem die Ableitung von $\nabla\tilde{\phi}$, i.Z. $\nabla^2\tilde{\phi}$, existiert. Wir geben im Folgenden die aus [25] abgeleitete Darstellung der zweiten Ableitung von $\tilde{\phi}$ an. Dazu benötigen wir die folgenden Definitionen:

Definition 6. *Sei $M \in \mathcal{S}^n$ gegeben. Dann exisitert eine Zerlegung $M = UDU^T$, wobei $U \in \mathbb{R}^{n\times n}$ eine orthogonale Matrix und D eine Diagonalmatrix ist, deren Diagonalelemente D_{ii} den Eigenwerten von M entsprechen. Die Betragsfunktion von M ist dann definiert durch*

$$|M| := U|D|U^T := U \begin{pmatrix} |D_{11}| & & \\ & \ddots & \\ & & |D_{nn}| \end{pmatrix} U^T.$$

Definition 7. *Sei $M \in \mathcal{S}^n$ gegeben. Die Funktion*

$$L_M : \mathcal{S}^n \to \mathcal{S}^n,\ L_M(X) = MX + XM$$

heißt Lyapunov-Operator (bzgl. M).

Lemma 8. *Sei $M \in \mathcal{S}^n_{++}$ mit der Eigenwertzerlegung $M = UDU^T$ gegeben. Dann ist L_M invertierbar und es gilt für ein $Y \in \mathcal{S}^n$*

$$L_M^{-1}(Y) = U\left[\left(\frac{(U^TYU)_{ij}}{D_{ii}+D_{jj}}\right)_{i,j=1\dots n}\right]U^T.$$

Beweis. Sei $Y = MX + XM$ für ein $X \in \mathcal{S}^n$. Die Eigenwertzerlegung ist $M = UDU^T$. Es folgt dann für $\tilde{X} := U^TXU$

$$\tilde{Y} := U^TYU = D\tilde{X} + \tilde{X}D = (\underbrace{(D_{ii}+D_{jj})}_{>0}\tilde{X}_{ij})_{i,j=1\dots n}.$$

Somit folgt $\tilde{X}_{ij} = \frac{\tilde{Y}_{ij}}{D_{ii}+D_{jj}}$ für $i,j = 1,\dots,n$. Damit ist $\tilde{X}$ eindeutig durch $\tilde{Y}$ bestimmt. Weil die Abbildung $\mathcal{U} : \mathcal{S}^n \to \mathcal{S}^n$, $\mathcal{U}(G) = U^TGU$, bijektiv ist, folgt die Invertierbarkeit von L_M. □

Die folgende Aussage wird bei der Untersuchung von $\nabla^2\tilde{\phi}$ von Bedeutung sein. Sie wurde in [8] gezeigt:

Lemma 9. *Die Abbildung $(Y, M) \mapsto L_M^{-1}(Y)$ ist in jedem Punkt $(Y, M) \in \mathcal{S}^n \times \mathcal{S}^n_{++}$ stetig.*

Es gilt $\Pi_{\mathcal{S}^n_+}(X) = \frac{1}{2}(X + |X|)$. Wir werden diesen Zusammenhang zunächst nutzen, um die Hessematrix von $\tilde{\phi}(X, S)$ für $(X, S) \in \mathbf{A}$ mit $\det(XS) \neq 0$ anzugeben. Dazu verwenden wir weitere Aussagen aus [25]:

Satz 5. *Sei $\vartheta(Y) := |Y|$ für $Y \in S^n$. Die Funktion ϑ ist lokal Lipschitz-stetig und für alle invertierbaren Y differenzierbar. Es gilt*

$$\nabla\vartheta(Y)[H] = L_{|Y|}^{-1}(YH + HY).$$

Daraus lässt sich die Jacobimatrix von ϑ an einer Stelle $Y \in \mathcal{S}^n$ bei Bedarf explizit ausrechnen, sofern Y invertierbar ist. Der folgende Satz aus [25] ist für die späteren Überlegungen ebenfalls von Bedeutung:

Satz 6. *Sei ϑ wie in Satz 5 definiert und ein $X \in S^n$ gegeben. Es existieren alle Richtungsableitungen $\vartheta'(X, H)$ an der Stelle X und es gilt für eine gegebene Eigenwertzerlegung*

$$X = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} U^T$$

mit $D \in \mathcal{S}^k$, $\det(D) \neq 0$,

$$\tilde{H} := \begin{pmatrix} \tilde{H}_{11} & \tilde{H}_{12} \\ \tilde{H}_{12}^T & \tilde{H}_{22} \end{pmatrix} := U^T H U$$

und $\bar{D} := \vartheta(D) = |D|$:

$$\vartheta'(X, H) = U \begin{pmatrix} L_{\bar{D}}^{-1}(D\tilde{H}_{11} + \tilde{H}_{11}D) & \bar{D}^{-1}D\tilde{H}_{12} \\ \tilde{H}_{12}^T D\bar{D}^{-1} & |\tilde{H}_{22}| \end{pmatrix} U^T.$$

Ist $X \in \mathcal{S}^n$ invertierbar, dann gilt natürlich $\vartheta'(X, H) = \nabla\vartheta(X)[H]$. Aus dem bereits oben erwähnten Zusammenhang $\Pi_{\mathcal{S}^n_+}(X) = \frac{1}{2}(X + |X|)$ und Satz 5 kann man für invertierbares $X \in \mathcal{S}^n$ die Ableitung von $\Pi_{\mathcal{S}^n_+}$ an der Stelle X mit $\nabla\Pi_{\mathcal{S}^n_+}(X)[H]$ beschreiben. Es gilt:

$$\nabla\Pi_{\mathcal{S}^n_+}(X)[H] = \frac{1}{2}(H + \nabla\vartheta(X)[H]).$$

Die folgende Herleitung von $\nabla^2\tilde{\phi}$ wurde in [8] angegeben:
Sei $\zeta : \mathcal{S}^n \to \mathcal{S}^n$, $\zeta(X) = X - \Pi_{\mathcal{S}^n_+}(X)$. Unter Verwendung von Satz 5 folgt für invertierbares $X \in \mathcal{S}^n$:

$$\nabla\zeta(X)[H] = H - \nabla\Pi_{\mathcal{S}^n_+}(X)[H] = \frac{1}{2}(H - L_{|X|}^{-1}(XH + HX)).$$

Da der Lyapunov-Operator linear ist, folgt weiterhin

$$\begin{aligned}\nabla\zeta(X)[H] &= \tfrac{1}{2}(H - L_{|X|}^{-1}(XH + HX)) \\ &= \tfrac{1}{2}(L_{|X|}^{-1}(L_{|X|}(H)) - L_{|X|}^{-1}(L_X(H)) \\ &= \tfrac{1}{2}(L_{|X|}^{-1}(L_{|X|}(H) - L_X(H)) \\ &= \tfrac{1}{2}L_{|X|}^{-1} \circ L_{|X|-X}(H).\end{aligned}$$

Für die in Abschnitt 2.2 auf $\mathcal{S}^n \times \mathcal{S}^n$ definierte Funktion χ erhalten wir damit für X, $S \in \mathcal{S}^n$ mit $\det(XS) \neq 0$

$$\begin{aligned}\nabla\chi(X,S)[H_1,H_2] &= (\nabla\zeta(X)[H_1], \nabla\zeta(S)[H_2]) \\ &= \tfrac{1}{2}(L_{|X|}^{-1} \circ L_{|X|-X}(H_1), L_{|S|}^{-1} \circ L_{|S|-S}(H_2))\end{aligned}$$

für H_1, $H_2 \in \mathcal{S}^n$. Wegen $\nabla\tilde{\phi}(X,S) = \Pi_{\mathbf{L}}(\chi(X,S))$ folgt für $(X,S) \in \mathbf{A}$ mit $\det(XS) \neq 0$

$$\nabla^2\tilde{\phi}(X,S)[(H_1,H_2)] = \frac{1}{2}\Pi_{\mathbf{L}}((L_{|X|}^{-1} \circ L_{|X|-X}(H_1), L_{|S|}^{-1} \circ L_{|S|-S}(H_2))) \qquad (2.2)$$

für $(H_1,H_2) \in \mathbf{L}$. Da die voneinander abhängigen Funktionen ϑ und $\Pi_{\mathcal{S}_+^n}$ lokal Lipschitz-steig sind (Satz 5 bzw. Satz 3), folgt aus der oben beschriebenen Zusammensetzung von $\nabla\tilde{\phi}$, dass $\nabla\tilde{\phi}$ auf $\mathbf{A}$ lokal Lipschitz-stetig und damit nach Rademachers Theorem fast überall auf $\mathbf{A}$ differenzierbar ist. Wir haben außerdem bereits festgestellt, dass die Nullmenge der nicht differenzierbaren Punkte, i.Z. $\overline{\mathcal{CS}}_{\tilde{\phi}}$, eine Teilmenge von $\mathcal{CS}_{\tilde{\phi}}$ ist. Diese Nullmenge kann bei iterativen Verfahren erster Ordnung zur Lösung der Probleme (P) und (D) vernachlässigt werden. Sollte das gewählte Verfahren auch die zweite Ableitung verwenden und ein Punkt $Z \in \mathcal{CS}_{\tilde{\phi}} \setminus \overline{\mathcal{CS}}_{\tilde{\phi}}$ bestimmt werden, dann kann $\nabla^2\tilde{\phi}(Z)$ mit Satz 6 berechnet werden. Wir werden darauf in Lemma 11 und bei der Untersuchung des Newton-Verfahrens zur Minimierung von $\tilde{\phi}$ noch eingehen.

Wir wissen bereits, dass eine Optimallösung (X^{opt}, S^{opt}) der Probleme (P) und (D) ein Punkt sein kann, an dem die zugehörige Funktion $\tilde{\phi}$ nicht zweimal differenzierbar ist. Es ist nun die Frage, ob man unter gewissen Voraussetzungen zeigen kann, dass die verallgemeinerte Hessematrix ausschließlich positiv definite Elemente enthält, d.h. $\partial\tilde{\phi}(X^{opt}, S^{opt}) \succ 0$. Dazu zeigen wir zunächst die folgende Aussage:

Lemma 10. *Die Funktion $\nabla\tilde{\phi}$ ist auf der Menge $\mathbf{A} \setminus \mathcal{CS}_{\tilde{\phi}}$ stetig differenzierbar.*

Beweis. In Abschnitt 2.2 sind wir bereits darauf eingegangen, dass $\nabla\tilde{\phi}$ auf der Menge $\mathbf{A} \setminus \mathcal{CS}_{\tilde{\phi}}$ differenzierbar ist. Sei nun ein beliebiger Punkt $(X,S) \in \mathbf{A} \setminus \mathcal{CS}_{\tilde{\phi}}$ gegeben. Aus der Differenzierbarkeit folgt die Existenz aller Richtungsableitungen und damit insbesondere aller partiellen Ableitungen. Wir betrachten nun (2.2). Aus Satz 5 folgt, dass die Betragsabbildung ϑ stetig ist. Damit ist auch die

Abbildung $G \mapsto |G| - G$ stetig. Aus Definition 7 folgt unmittelbar, dass die Abbildung $(Y, M) \mapsto L_M(Y)$ stetig ist. Es folgt somit, dass $X \mapsto L_{|X|-X}(H_1)$ stetig ist. Da X invertierbar ist folgt $|X| \succ 0$ und deshalb ist nach Lemma 8 auch $X \mapsto L^{-1}_{|X|}(L_{|X|-X}(H_1))$ stetig. Analog für S. Da $\frac{1}{2}\Pi_{\mathbf{L}}$ eine lineare Abbildung ist, folgt insgesamt, dass $(X, S) \mapsto \nabla^2\tilde{\phi}(X, S)[(H_1, H_2)]$ stetig ist. Damit ist jede Richtungs- und jede partielle Ableitung stetig. Somit folgt, dass $\nabla^2\tilde{\phi}$ an der Stelle (X, S) stetig differenzierbar ist. □

Da die Menge $\mathbf{A} \setminus \mathcal{CS}_{\tilde{\phi}}$ in $\mathbf{A}$ offen ist, folgt die Symmetrie und wegen der Konvexität von $\tilde{\phi}$ auf $\mathbf{A}$ auch die positive Semidefinitheit der Bilinearform $\nabla^2\tilde{\phi}(X, S)$ für alle $(X, S) \in \mathbf{A} \setminus \mathcal{CS}_{\tilde{\phi}}$.

Als Nächstes befassen wir uns mit der Menge $\mathcal{CS}_{\tilde{\phi}}$.

Lemma 11. *Ist die Fuktion $\nabla\tilde{\phi}$ an einer Stelle $(X, S) \in \mathcal{CS}_{\tilde{\phi}}$ differenzierbar, dann gilt $\nabla^2\tilde{\phi}(X, S) \succeq 0$.*

Beweis. Wir verwenden die Richtungsableitung der Betragsfunktion aus Satz 6 und deren Zusammenhang zur Projektion $\Pi_{\mathcal{S}^n_+}$: Sei $(X, S) \in \mathcal{CS}_{\tilde{\phi}}$ gegeben. Ist $\nabla\tilde{\phi}$ an der Stelle (X, S) differenzierbar, dann gelten die folgenden Aussagen:
Sei für X und für S die Eigenwertzerlegung

$$X = U \begin{pmatrix} P & 0 \\ 0 & 0 \end{pmatrix} U^T,$$
$$S = V \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix} V^T,$$

mit $P \in \mathcal{S}^k$, $Q \in \mathcal{S}^l$, $\det(P) \neq 0 \neq \det(Q)$, und für ein $(H, G) \in \mathbf{L}$

$$\tilde{H} := \begin{pmatrix} \tilde{H}_{11} & \tilde{H}_{12} \\ \tilde{H}_{12}^T & \tilde{H}_{22} \end{pmatrix} := U^T H U,$$
$$\tilde{G} := \begin{pmatrix} \tilde{G}_{11} & \tilde{G}_{12} \\ \tilde{G}_{12}^T & \tilde{G}_{22} \end{pmatrix} := V^T G V$$

gegeben. Mit $\bar{P} := \vartheta(P) = |P|$ und $\bar{Q} := \vartheta(Q) = |Q|$ folgt aus Satz 6 und der Differenzierbarkeit von $\nabla\tilde{\phi}$:

$$\begin{aligned}
&\chi'((X, S), (H, G)) \\
&= \begin{pmatrix} H - \Pi'_{\mathcal{S}^n_+}(X, H) \\ G - \Pi'_{\mathcal{S}^n_+}(S, G) \end{pmatrix} = \tfrac{1}{2} \begin{pmatrix} H + \vartheta'(X, H) \\ G + \vartheta'(S, G) \end{pmatrix} \\
&= \tfrac{1}{2} \begin{pmatrix} U\tilde{H}U^T + U \begin{bmatrix} L^{-1}_{\bar{P}}(P\tilde{H}_{11} + \tilde{H}_{11}P) & \bar{P}^{-1}P\tilde{H}_{12} \\ \tilde{H}_{12}^T P\bar{P}^{-1} & |\tilde{H}_{22}| \end{bmatrix} U^T \\ \\ V\tilde{G}V^T + V \begin{bmatrix} L^{-1}_{\bar{Q}}(Q\tilde{G}_{11} + \tilde{G}_{11}Q) & \bar{Q}^{-1}Q\tilde{G}_{12} \\ \tilde{G}_{12}^T Q\bar{Q}^{-1} & |\tilde{G}_{22}| \end{bmatrix} V^T \end{pmatrix}
\end{aligned}$$

und weiterhin

$$\nabla^2\tilde{\phi}(X,S)[(H,G)] = \Pi_{\mathbf{L}}(\chi'((X,S),(H,G))).$$

Da $\nabla^2\tilde{\phi}(X,S)[(H,G)] = -\nabla^2\tilde{\phi}(X,S)[-(H,G)]$ gilt, folgt insbesondere

$$\Pi_{\mathbf{L}}(\chi'((X,S),(H,G))) = \Pi_{\mathbf{L}}(-\chi'((X,S),-(H,G))). \tag{2.3}$$

Damit erhalten wir mit den Eigenschaften des spur- und des Lyapunov-Operators

$$\begin{aligned}
&2\nabla^2\tilde{\phi}(X,S)[(H,G),(H,G)]\\
&= 2\langle (H,G), \nabla^2\tilde{\phi}(X,S)[(H,G)]\rangle_{\bar{E}}\\
&= 2\langle (H,G), \Pi_{\mathbf{L}}(\chi'((X,S),(H,G)))\rangle_{\bar{E}}\\
&= 2\langle \Pi_{\mathbf{L}}(H,G), \chi'((X,S),(H,G))\rangle_{\bar{E}}\\
&= H \bullet (H + \vartheta'(X,H)) + G \bullet (G + \vartheta'(S,G))\\
&= \mathrm{spur}(\tilde{H}\tilde{H}) + \mathrm{spur}(\tilde{H}_{11}L_{\bar{P}}^{-1}(P\tilde{H}_{11} + \tilde{H}_{11}P)) + 2\mathrm{spur}(\tilde{H}_{12}\tilde{H}_{12}^T P\bar{P}^{-1})\\
&\quad + \mathrm{spur}(\tilde{G}\tilde{G}) + \mathrm{spur}(\tilde{G}_{11}L_{\bar{Q}}^{-1}(Q\tilde{G}_{11} + \tilde{G}_{11}Q)) + 2\mathrm{spur}(\tilde{G}_{12}\tilde{G}_{12}^T Q\bar{Q}^{-1})\\
&\quad + \underbrace{\mathrm{spur}(\tilde{H}_{22}|\tilde{H}_{22}|) + \mathrm{spur}(\tilde{G}_{22}|\tilde{G}_{22}|)}_{=-(\mathrm{spur}(\tilde{H}_{22}|\tilde{H}_{22}|)+\mathrm{spur}(\tilde{G}_{22}|\tilde{G}_{22}|))=0 \text{ nach (2.3)}}\\
&= \textstyle\sum_{i,j=1}^n \tilde{H}_{(i,j)}^2 + \mathrm{spur}(\tilde{H}_{11}(\frac{P_{(i,i)}+P_{(j,j)}}{\bar{P}_{(i,i)}+\bar{P}_{(j,j)}}\tilde{H}_{(i,j)})_{i,j=1}^k) + 2\sum_{\substack{i=1,\dots,k\\ j=k+1,\dots,n}} (\frac{P_{(i,i)}}{\bar{P}_{(i,i)}}\tilde{H}_{(i,j)}^2)\\
&\quad + \textstyle\sum_{i,j=1}^n \tilde{G}_{(i,j)}^2 + \mathrm{spur}(\tilde{G}_{11}(\frac{Q_{(i,i)}+Q_{(j,j)}}{\bar{Q}_{(i,i)}+\bar{Q}_{(j,j)}}\tilde{G}_{(i,j)})_{i,j=1}^l) + 2\sum_{\substack{i=1,\dots,l\\ j=l+1,\dots,n}} (\frac{Q_{(i,i)}}{\bar{Q}_{(i,i)}}\tilde{G}_{(i,j)}^2)\\
&= \textstyle\sum_{i,j=1}^n \tilde{H}_{(i,j)}^2 + \sum_{i,j=1}^k (\underbrace{\frac{P_{(i,i)}+P_{(j,j)}}{\bar{P}_{(i,i)}+\bar{P}_{(j,j)}}}_{\in[-1,1]}\tilde{H}_{(i,j)}^2) + 2\sum_{\substack{i=1,\dots,k\\ j=k+1,\dots,n}} (\underbrace{\frac{P_{(i,i)}}{\bar{P}_{(i,i)}}}_{\in\{-1,1\}}\tilde{H}_{(i,j)}^2)\\
&\quad + \textstyle\sum_{i,j=1}^n \tilde{G}_{(i,j)}^2 + \sum_{i,j=1}^l (\underbrace{\frac{Q_{(i,i)}+Q_{(j,j)}}{\bar{Q}_{(i,i)}+\bar{Q}_{(j,j)}}}_{\in[-1,1]}\tilde{G}_{(i,j)}) + 2\sum_{\substack{i=1,\dots,l\\ j=l+1,\dots,n}} (\underbrace{\frac{Q_{(i,i)}}{\bar{Q}_{(i,i)}}}_{\in\{-1,1\}}\tilde{G}_{(i,j)}^2)\\
&\geq 0.
\end{aligned}$$

Wir zeigen noch, dass $\nabla^2\tilde{\phi}(X,S)$ symmetrisch ist: Seien $(H,G),(R,T) \in \mathbf{L}$ gegeben. Es folgt

$$\begin{aligned}
&2\nabla^2\tilde{\phi}(X,S)[(H,G),(R,T)]\\
&= \mathrm{spur}(\tilde{H}\tilde{R}) + \mathrm{spur}(\tilde{H}_{11}L_{\bar{P}}^{-1}(P\tilde{R}_{11} + \tilde{R}_{11}P)) + 2\mathrm{spur}(\tilde{H}_{12}\tilde{R}_{12}^T P\bar{P}^{-1})\\
&\quad + \mathrm{spur}(\tilde{G}\tilde{T}) + \mathrm{spur}(\tilde{G}_{11}L_{\bar{Q}}^{-1}(Q\tilde{T}_{11} + \tilde{T}_{11}Q)) + 2\mathrm{spur}(\tilde{G}_{12}\tilde{T}_{12}^T Q\bar{Q}^{-1})\\
&\quad + \underbrace{\mathrm{spur}(\tilde{H}_{22}|\tilde{R}_{22}|) + \mathrm{spur}(\tilde{G}_{22}|\tilde{T}_{22}|)}_{=-(\mathrm{spur}(\tilde{H}_{22}|\tilde{R}_{22}|)+\mathrm{spur}(\tilde{G}_{22}|\tilde{T}_{22}|))=0 \text{ nach (2.3)}}
\end{aligned}$$

$$
\begin{aligned}
&= \operatorname{spur}(\tilde{R}\tilde{H}) + \textstyle\sum_{i,j=1}^{k}(\frac{P_{(i,i)}+P_{(j,j)}}{\bar{P}_{(i,i)}+\bar{P}_{(j,j)}}\tilde{H}_{(i,j)}\tilde{R}_{(i,j)}) + 2\operatorname{spur}(\tilde{R}_{12}\tilde{H}_{12}^T P\bar{P}^{-1}) \\
&\quad + \operatorname{spur}(\tilde{T}\tilde{G}) + \textstyle\sum_{i,j=1}^{l}(\frac{Q_{(i,i)}+Q_{(j,j)}}{\bar{Q}_{(i,i)}+\bar{Q}_{(j,j)}}\tilde{G}_{(i,j)}\tilde{T}_{(i,j)}) + 2\operatorname{spur}(\tilde{T}_{12}\tilde{G}_{12}^T Q\bar{Q}^{-1}) \\
&\quad + \underbrace{\operatorname{spur}(\tilde{R}_{22}|\tilde{H}_{22}|) + \operatorname{spur}(\tilde{T}_{22}|\tilde{G}_{22}|)}_{=-(\operatorname{spur}(\tilde{R}_{22}|\tilde{H}_{22}|)+\operatorname{spur}(\tilde{T}_{22}|\tilde{G}_{22}|))=0 \text{ nach (2.3)}} \\
&= 2\nabla^2\tilde{\phi}(X,S)[(R,T),(H,G)].
\end{aligned}
$$

Insgesamt haben wir somit gezeigt, dass $\nabla^2\tilde{\phi}(X,S)$ symmetrisch und positiv semidefinit ist. □

Korollar 3. *Sei* $\operatorname{Diff}(\nabla\tilde{\phi}) \subseteq \mathbf{A}$ *der Definitionsbereich von* $\nabla^2\tilde{\phi}$*. Es existiert eine von der entsprechenden Norm* $\|.\|$ *abhängige Konstante* $K_{\|.\|}$*, so dass für alle* $Z \in \operatorname{Diff}(\nabla\tilde{\phi})$

$$\|\nabla^2\tilde{\phi}(Z)\| \leq K_{\|.\|}$$

gilt. Weiterhin gilt für alle $Z \in \mathbf{A}$

$$\sup_{M\in\partial^2\tilde{\phi}(Z)} \|M\| \leq K_{\|.\|}$$

Beweis. Aus dem Beweis von Lemma 11 folgt für $Z \in \mathcal{CS}_{\tilde{\phi}} \cap \operatorname{Diff}(\nabla\tilde{\phi})$ und analog für $Z \in \mathbf{A} \setminus \mathcal{CS}_{\tilde{\phi}} \subseteq \operatorname{Diff}(\nabla\tilde{\phi})$

$$\nabla^2\tilde{\phi}(Z)[(H,G),(H,G)] \leq H \bullet H + G \bullet G = \|(H,G)\|_{\bar{F}}^2.$$

Daraus folgt sofort die erste Ungleichung. Aus der Definition von $\partial^2\tilde{\phi}(Z)$ und den Normeigenschaften folgt somit für alle $Z \in \mathbf{A}$, dass die Norm jedes Elementes $M \in \partial^2\tilde{\phi}(Z)$ mit derselben Konstante abgeschätzt werden kann. □

Satz 7. *Seien konische Programme der Form* (P) *und* (D) *gegeben. Weiterhin sei Voraussetzung 1 erfüllt. Für alle* $Z \in \mathbf{A}$ *gilt*

$$\partial^2\tilde{\phi}(Z) \succeq 0.$$

Beweis. Nach Lemma 10 und 11 ist die Abbildung $\nabla^2\tilde{\phi}$ auf ihrem Definitionsbereich $\operatorname{Diff}(\nabla\tilde{\phi})$ symmetrisch und positiv semidefinit. Sei eine Folge $Z^k \to Z$ mit $Z^k \in \operatorname{Diff}(\nabla\tilde{\phi})$ für alle $k \in \mathbb{N}$ und der folgenden Eigenschaft gegeben: Es existiert eine Bilinearform $BF_{(Z^k)} : \mathbf{L} \times \mathbf{L} \to I\!R$, für welche $\nabla^2\tilde{\phi}(Z^k) \to BF_{(Z^k)}$ gilt. Wegen $BF_{(Z^k)}[G,H] \leftarrow \nabla^2\tilde{\phi}(Z^k)[G,H] = \nabla^2\tilde{\phi}(Z^k)[H,G] \to BF_{(Z^k)}[H,G]$ und $0 \leq \nabla^2\tilde{\phi}(Z^k)[H,H] \to BF_{(Z^k)}[H,H]$ für alle G, $H \in \mathbf{L}$ ist $BF_{(Z^k)}$ symmetrisch und positiv semidefinit. Da die Elemente in der konvexen Hülle aller Bilinearformen $BF_{(Z^k)}$ ebenfalls symmetrisch und positiv semidefinit sind, folgt die Aussage. □

Satz 7 gilt insbesondere für jede Optimallösung Z^{opt} eines gegebenen primal-dualen Paares. Damit die Voraussetzungen von Satz 4 erfüllt sind, müsste aber $\partial^2\tilde{\phi}(Z^{opt}) \succ 0$ gelten. Das dies i.Allg. nicht der Fall ist, zeigt das folgende Beispiel aus [9]:

Beispiel 1. *Seien die konischen Programme (P) und (D) mit den folgenden Daten gegeben:*

$$C = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},\ B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

$$\mathcal{L} = \{X \mid A^{(1)} \bullet X = 0\} \text{ mit } A^{(1)} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Da $B \bullet C = 0$ und $B, C \succeq 0$ ist $(X^{opt}, S^{opt}) := (B, C)$ nach Korollar 1 eine Optimallösung von (P) und (D). Wir merken weiter an, dass wegen

$$\begin{pmatrix} 3 & -1 \\ -1 & 1 \end{pmatrix} \in (\mathcal{L} + B) \cap \mathcal{S}^n_{++},\ \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \in (\mathcal{L}^\perp + C) \cap \mathcal{S}^n_{++}$$

Voraussetzung 2 erfüllt ist.
Man rechnet leicht nach, dass $\mathbf{L}$ folgende Gestalt hat:

$$\mathbf{L} = \left\{ \left(\begin{pmatrix} 2a & -a \\ -a & b \end{pmatrix}, \begin{pmatrix} -b & -b \\ -b & 0 \end{pmatrix} \right) \mid a, b \in I\!R \right\}.$$

Wir wählen die Richtung

$$(\Delta X, \Delta S) := \left(\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & -1 \\ -1 & 0 \end{pmatrix} \right) \in \mathbf{L}$$

und definieren $(X(t), S(t)) := (X^{opt}, S^{opt}) + t(\Delta X, \Delta S)$. Für alle $t \in (0, 1)$ gilt $X(t) \in \mathcal{S}^n_{++}$ und es existiert $S(t)^{-1}$. Für $t \searrow 0$ konvergiert $(X(t), S(t))$ gegen (X^{opt}, S^{opt}) und für

$$H := \left(\begin{pmatrix} 2 & -1 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \right) \in \mathbf{L}$$

folgt für alle $t \in (0, 1)$

$$\tilde{\phi}((X(t), S(t)) + \alpha H) = const(t)\ \forall \alpha \in [\max(-\frac{1}{2}, t - \sqrt{t^2 + t}), t + \sqrt{t^2 + t}].$$

Sei nun eine Folge $t_k \searrow 0$ mit $t_k < 1\ \forall k \in \mathbb{N}$ gegeben. Mit $(X(t_k), S(t_k)) \in \mathbf{A} \setminus \mathcal{CS}_{\tilde{\phi}}$ folgt, dass $\nabla^2\tilde{\phi}(X(t_k), S(t_k))$ existiert. Aus dem eben gezeigten folgt weiterhin, dass $\nabla^2\tilde{\phi}(X(t_k), S(t_k))[H, H] = 0$ gilt. Sei oBdA die Folge $(t_k)_{k\in\mathbb{N}}$ so gewählt, dass $M := \lim_{k\to\infty} \nabla^2\tilde{\phi}(X(t_k), S(t_k))$ existiert. (Die Existenz einer solchen Folge wird durch Korollar 3 gewährleistet.) Die Bilinearform M erfüllt dann $M[H, H] = 0$ und ist in $\partial^2\tilde{\phi}(X^{opt}, S^{opt})$ enthalten.

Dieses Beispiel zeigt, dass die Voraussetzungen von Satz 4 für die zu einem primal-dualen Paar (P) und (D) gehörenden Daten $\mathbf{A}$, $\mathbf{K}$ und $\tilde{\phi}$ verletzt sein können. In einem solchen Fall kann die Konvergenzgeschwindigkeit des verallgemeinerten Newton-Verfahrens beliebig schlecht werden, falls es überhaupt anwendbar ist.

2.4 Regularisierungen

Um die Konvergenzgeschwindigkeit der APD-Methode zu erhöhen, falls die Suchschritte mit dem Newton- oder einem Quasi-Newton- Verfahren berechnet werden, wurde in [9] eine Variante vorgeschlagen, welche wir im Folgenden beschreiben werden:

Wir zitieren zunächst eine Aussage aus [20]. Seien ein euklidischer Raum U und eine Funktion $\eta : U \to U$ gegeben.

Definition 8. *Für ein $u \in U$ exisitere eine Abbildung $B\eta(u) : U \to U$ mit den Eigenschaften*
i) $B\eta(u)[tv] = tB\eta(u)[v]$ für alle $t \geq 0$, $v \in U$,
ii)

$$\lim_{v \to 0} \frac{\|\eta(u+v) - \eta(u) - B\eta(u)[v]\|_2}{\|v\|_2} = 0.$$

Dann heißt $B\eta(u)$ die B-Ableitung von η in u.

Lemma 12. *Ist die B-Ableitung $B\eta$ lokal Lipschitz-stetig in $u \in U$, dann ist η lokal Lipschitz-stetig und weiterhin stark semiglatt in u.*

Als Nächstes betrachten wir die Funktion

$$\begin{array}{rccl} \bar{f}: & \mathcal{S}^n \times \mathcal{S}^n & \to & I\!R, \\ & (X, S) & \mapsto & \|XS - SX\|_F^2. \end{array}$$

Die Funktion $\bar{f}$ ist beliebig oft differenzierbar und es gilt $\bar{f}(X^{opt}, S^{opt}) = 0$, falls (X^{opt}, S^{opt}) die Optimallösung eines primal-dualen Paares (P) und (D) ist, siehe (2.1). Da $\bar{f} \geq 0$ gilt, ist jede Optimallösung (X^{opt}, S^{opt}) Minimalstelle der Funktion $\bar{f}$. Wir setzen $\hat{f} := \bar{f}_{|\mathbf{A}}$. (X^{opt}, S^{opt}) ist dann auch Minimalstelle der auf $\mathbf{A}$ beliebig oft differenzierbaren Funktion $\hat{f}$. Somit gilt $\nabla^2 \hat{f}(X^{opt}, S^{opt}) \succeq 0$.

Lemma 13. *Die Funktion $\nabla \hat{f}$ ist lokal Lipschitz-stetig und stark semiglatt auf dem affinen Raum $\mathbf{A}$.*

Beweis. Sei $Z \in \mathbf{A}$ gegeben. Da $\nabla \hat{f}$ in Z differenzierbar ist, ist $\nabla^2 \hat{f}(Z)$ die B-Ableitung von $\nabla \hat{f}$ in Z. Da $\nabla^2 \hat{f}$ stetig differenzierbar ist, ist diese Abbildung auch lokal Lipschitz-stetig in Z. Nach Lemma 12 ist $\nabla \hat{f}$ damit lokal Lipschitz-stetig und stark semiglatt in Z. □

Satz 8. *Die Funktion $\nabla(\tilde{\phi}+\hat{f})$ ist lokal Lipschiz-stetig, stark semiglatt und fast überall differenzierbar auf* $\mathbf{A}$. *Es gilt $\partial^2(\tilde{\phi}+\hat{f})(X^{opt},S^{opt}) \succeq 0$.*

Beweis. Nach Korollar 2 ist $\nabla\tilde{\phi}$ lokal Lipschitz-stetig, stark semiglatt und fast überall differenzierbar auf $\mathbf{A}$. Da sich die lokale Lipschitz-Stetigkeit bei Summenbildung überträgt, gilt diese Eigenschaft nach Lemma 13 auch für $\nabla\tilde{\phi}+\nabla\hat{f}$. Genauso verhält es sich bei der Differenzierbarkeit. Wir betrachten noch die starke Semiglattheit:
Aus Satz 6 und dem Beweis von Lemma 11 folgt, dass alle Richtungsableitungen von $\nabla\tilde{\phi}$ und damit auch von $\nabla\tilde{\phi}+\nabla\hat{f}$ in jedem $Z \in \mathbf{A}$ existieren. Da $\nabla^2\hat{f}$ stetig ist, folgt für ein $Z \in \mathbf{A}$ aus Lemma 4 $\partial^2\hat{f}(Z) = \{\nabla^2\hat{f}(Z)\}$ und somit

$$\partial^2(\tilde{\phi}+\hat{f})(Z) = \partial^2\tilde{\phi}(Z) + \nabla^2\hat{f}(Z). \tag{2.4}$$

Es folgt damit unter Verwendung von Korollar 2 und Lemma 13:
Sei $\tilde{Z} \to Z$ und $M_{\tilde{Z}} \in \partial^2(\tilde{\phi}+\hat{f})(\tilde{Z})$ gegeben. Dann gilt $M_{\tilde{Z}} = \tilde{M}_{\tilde{Z}} + \nabla^2\hat{f}(\tilde{Z})$ mit $\tilde{M}_{\tilde{Z}} \in \partial^2\tilde{\phi}(\tilde{Z})$ und somit

$$\begin{aligned}
&\|\nabla(\tilde{\phi}+\hat{f})(\tilde{Z}) - \nabla(\tilde{\phi}+\hat{f})(Z) - M_{\tilde{Z}}(\tilde{Z}-Z)\|_{\bar{F}} \\
&\leq \|\nabla\tilde{\phi}(\tilde{Z}) - \nabla\tilde{\phi}(Z) - \tilde{M}_{\tilde{Z}}(\tilde{Z}-Z)\|_{\bar{F}} + \|\nabla\hat{f}(\tilde{Z}) - \nabla\hat{f}(Z) - \nabla^2\hat{f}(\tilde{Z})(\tilde{Z}-Z)\|_{\bar{F}} \\
&= \mathcal{O}(\|\tilde{Z}-Z\|_{\bar{F}}^2) + \mathcal{O}(\|\tilde{Z}-Z\|_{\bar{F}}^2).
\end{aligned}$$

Aus Satz 7, $\nabla^2\hat{f}(X^{opt},S^{opt}) \succeq 0$ und (2.4) folgt schließlich $\partial^2(\tilde{\phi}+\hat{f})(X^{opt},S^{opt}) \succeq 0$. □

Damit besitzt $\nabla(\tilde{\phi}+\hat{f})$ die gleichen Eigenschaften wie $\nabla\tilde{\phi}$. In [9] wurde außerdem die folgende Eigenschaft für $\tilde{\phi}+\hat{f}$ gezeigt:

Voraussetzung 3. *Seien Programme (P) und (D) gegeben, die Voraussetzung 1 erfüllen. Die Optimallösung $Z^{opt} = (X^{opt},S^{opt})$ sei eindeutig und strikt komplementär, d.h. es gelte $X^{opt}+S^{opt} \succ 0$.*

Satz 9. *Sei für (P) und (D) Voraussetzung 3 erfüllt. Es existiert ein $\rho > 0$, so dass*

$$\tilde{\phi}(Z^{opt}+\lambda\Delta Z) + \hat{f}(Z^{opt}+\lambda\Delta Z) \geq \rho\lambda^2 \tag{2.5}$$

für alle $\Delta Z = (\Delta X,\Delta S) \in \mathbf{L}$ mit $\|\Delta X\|_F^2 + \|\Delta S\|_F^2 = 1$ erfüllt ist, wobei $\lambda \in [0,\tilde{\lambda}(\Delta Z)]$ mit $\tilde{\lambda}(\Delta Z) > 0$ gilt.

Man bezeichnet Eigenschaft (2.5) allgemein als Wachstumsbedingung 2. Ordnung (im Folgenden: WB2).
Wir zeigen noch eine Aussage, die unter etwas stärkeren Voraussetzungen gilt.

Lemma 14. *Sei U ein euklidischer Raum und $\eta : U \to \mathbb{R}$ eine Funktion, die auf einer Umgebung $B(u)$ von $u \in U$ zweimal differenzierbar ist. Sei $\nabla^2\eta$ auf dieser Umgebung beschränkt. Weiterhin gelte $\nabla\eta(u) = 0$ und für ein $\rho > 0$*

$$\eta(u + \lambda\Delta u) - \eta(u) \geq \rho\lambda^2$$

für alle $\Delta u \in U$ mit $\|\Delta u\|_2 = 1$ und alle $\lambda \in [0, \lambda(\Delta u)]$, $\lambda(\Delta u) > 0$. Es gilt dann $\nabla^2\eta(u) \succ 0$.

Beweis. Da η in u zweimal differenzierbar ist, ist $\nabla^2\eta(u)$ symmetrisch. Es folgt nun mit der Taylor-Entwicklung für ein beliebiges Δu mit $\|\Delta u\|_2 =: \delta \in (0, 1]$ und $\lambda \in [0, \frac{1}{\delta}\lambda(\frac{\Delta u}{\delta})]$ mit $u + \lambda\Delta u \in B(u)$:

$$\rho\lambda^2 \leq \eta(u + \lambda\Delta u) - \eta(u) \leq \lambda \underbrace{\langle\nabla\eta(u), \Delta u\rangle}_{=0} + 0.5\lambda^2\langle\Delta u, \nabla^2\eta(u)[\Delta u]\rangle + \lambda^2 const\delta^2.$$

Wähle δ so klein, dass $\rho - const\delta^2 > 0$ gilt. Es folgt durch Division mit λ^2, wobei $\lambda \in (0, \frac{1}{\delta}\lambda(\frac{\Delta u}{\delta})]$ sowie $u + \lambda\Delta u \in B(u)$ gelte:

$$0 < \rho - const\delta^2 \leq 0.5\langle\Delta u, \nabla^2\eta(u)[\Delta u]\rangle.$$

Insgesamt folgt somit $\langle\Delta u, \nabla^2\eta(u)[\Delta u]\rangle > 0$ für alle $\Delta u \neq 0$. □

Die Funktion $\tilde{\phi} + \hat{f}$ ist zwar an der Minimalstelle möglicherweise nicht zweimal differenzierbar, aber sie besitzt die Eigenschaften aus Satz 8 und unter Voraussetzung 3 auch die Eigenschaften aus Satz 9. Eine ähnliche Aussage wie in Lemma 14 erwartend (und deren Auswirkungen auf $\partial^2(\tilde{\phi} + \hat{f})(Z^{opt})$) wurde in [9] folgendes Vorgehen vorgeschlagen:

1. Minimiere $\tilde{\phi}$ um sich der Optimallösung ausreichend anzunähern.
2. Minimiere danach $\tilde{\phi} + \hat{f}$ um die Optimallösung zu bestimmen.

Im Allgemeinen sind die Eigenschaften aus Satz 8 und WG2 an einer Stelle u im Definitionsbereich einer stetig differenzierbaren Funktion η allerdings nicht hinreichend um $\partial^2\eta(u) \succ 0$ zu garantieren:

Beispiel 2. *Wir definieren die Funktion $\gamma : \mathbb{R}^2 \to \mathbb{R}$ durch*

$$\gamma(x, y) = \begin{cases} x^2 & \text{falls } x \geq |y|, \\ x^2 + (y - x)^2 & \text{falls } y > |x|, \\ x^2 + (y + x)^2 & \text{falls } y < -|x|, \\ 3x^2 + 2y^2 & \text{falls } x \leq -|y| \quad ((x, y) \neq (0, 0)). \end{cases}$$

Jede Teilfunktion ist auf dem Abschluss des entsprechenden Teilbereiches konvex und beliebig oft stetig differenzierbar. Man prüft leicht nach, dass γ auf $\mathbb{R}^2$ stetig

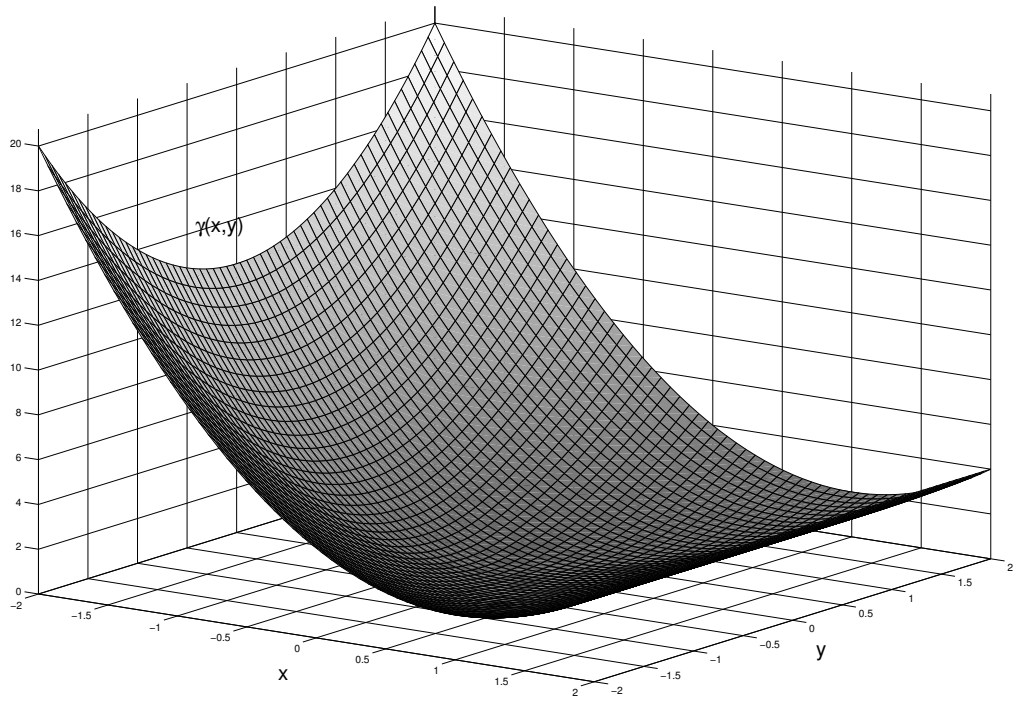

Abbildung 2.1: Die Funktion γ

ist. Da an jeder Randstelle eines Teilbereichs der Grenzwert aller Ableitungsfolgen (der Teilfunktionen) existiert und eindeutig ist, folgt die stetige Differenzierbarkeit von γ auf $\mathbb{R}^2$. Daraus ergibt sich, dass γ auf $\mathbb{R}^2$ konvex ist. Die eindeutige Minimalstelle ist der Punkt $(x^*, y^*) = (0, 0)$.
Die Ableitung von γ ist durch

$$\nabla\gamma(x,y) = \begin{cases} (2x,0)^T & \text{falls } x \geq |y|, \\ (4x-2y, 2y-2x)^T & \text{falls } y > |x|, \\ (4x+2y, 2y+2x)^T & \text{falls } y < -|x|, \\ (6x,4y)^T & \text{falls } x \leq -|y| \quad ((x,y) \neq (0,0)). \end{cases}$$

gegeben.
Die Funktion $\nabla\gamma$ ist stückweise linear und lokal Lipschitz-stetig.

$\nabla\gamma$ ist weiterhin stark semiglatt: Für das Innere der vier Teilbereiche folgt dies direkt aus Lemma 5. Um dies für die Randbereiche einzusehen betrachten wir zuerst die zweite Ableitung von γ. Diese ist auf dem Inneren aller vier Teilbereiche

definiert.

$$\nabla^2\gamma(x,y) = \begin{cases} \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} & \text{falls } x > |y|, \\ \begin{pmatrix} 4 & -2 \\ -2 & 2 \end{pmatrix} & \text{falls } y > |x|, \\ \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix} & \text{falls } y < -|x|, \\ \begin{pmatrix} 6 & 0 \\ 0 & 4 \end{pmatrix} & \text{falls } x < -|y|. \end{cases}$$

Die verallgemeinerte Hessematrix von γ an jeder beliebigen Stelle im $\mathbb{R}^2$ ist eine Teilmenge der konvexen Hülle dieser vier Matrizen.
Man prüft leicht nach, dass in jedem Punkt alle Richtungsableitungen von $\nabla\gamma$ existieren.
Sei $(w_k)_{k\in\mathbb{N}}$ eine Folge, die gegen $w \in \mathbb{R}^2$ konvergiert. Ist k hinreichend groß, dann befindet sich die gesamte Verbindungsstrecke von w_k nach w auf dem Rand von höchstens zwei benachbarten Teilbereichen (sofern der Fall $w_k = w = 0$ nicht vorliegt), auf denen die entsprechende Hessematrix konstant ist. Die konvexe Hülle dieser Matrizen (bzw. dieser Matrix) ist $\partial^2\gamma(w_k)$. Da $\nabla\gamma$ stetig und in jedem Teilbereich linear ist folgt dann für alle w_k, die nah genug bei w liegen, $\nabla\gamma(w_k) - \nabla\gamma(w) = M(w_k - w)$ für alle $M \in \partial^2\gamma(w_k)$ und somit die starke Semiglattheit von $\nabla\gamma$.

Da $\gamma(x,y) \geq \frac{1}{4}(x^2+y^2)\ \forall (x,y) \in \mathbb{R}^2$, ist WG2 in (x^*,y^*) erfüllt.

Für $x > |y|$ gilt jedoch

$$\nabla^2\gamma(x,y) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.6}$$

Daraus folgt direkt, dass $\partial^2\gamma(x^*,y^*)$ nicht invertierbare Matrizen enthält.
Die Funktion $\gamma(x,y) - \frac{1}{8}(x^2+y^2)$ besitzt eine lokal Lipschitz-stetige und stark semiglatte Ableitung und erfüllt ebenfalls WG2 in (x^*,y^*), obwohl sogar

$$\frac{1}{4}\begin{pmatrix} 7 & 0 \\ 0 & -1 \end{pmatrix} \in \partial^2\gamma(x^*,y^*) \tag{2.7}$$

gilt.

Die zu (P) und (D) gehörenden Räume $\mathbf{A}$ und $\mathbf{L}$ sind von einer besonderen Form, insbesondere falls Voraussetzung 3 erfüllt ist. Daher ist zunächst nicht offensichtlich, ob $\partial^2(\tilde{\phi}+\hat{f})(Z^{opt}) \not\succ 0$ wirklich eintreten kann. In Satz 8 haben wir bereits $\partial^2(\tilde{\phi}+\hat{f})(Z^{opt}) \succeq 0$ gezeigt. Somit kann ein zu (2.7) vergleichbarer Fall nicht vorkommen. Im folgenden Beispiel zeigen wir jedoch, dass ein Fall wie in (2.6) tatsächlich eintreten kann.

Beispiel 3. *Sei*

$$C := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \bar{b} := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

und

$$\mathcal{A}(X) = \begin{pmatrix} A^{(1)} \bullet X \\ A^{(2)} \bullet X \\ A^{(3)} \bullet X \end{pmatrix}$$

mit

$$A^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; A^{(2)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; A^{(3)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Wir betrachten das zugehörige semidefinite Programm $(\bar{P})$

$$min \;\; C \bullet X \quad | \quad \mathcal{A}(X) = \bar{b}, \;\; X \succeq 0.$$

Dieses Programm kann in die primal-duale Form (P) und (D) umgeschrieben werden (siehe Abschnitt 1.1). $\mathcal{L}$ ist somit der Nullraum von $\mathcal{A}$ und B eine symmetrische Matrix, die $\mathcal{A}(B) = \bar{b}$ erfüllt. Das primal-duale Paar erfüllt die Slater-Bedingung:

$$\tilde{X} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \in (\mathcal{L}+B) \cap \mathcal{S}^n_{++}, \; \tilde{S} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \in (\mathcal{L}^\perp + C) \cap \mathcal{S}^n_{++}.$$

Damit besitzt es eine Optimallösung (X^{opt}, S^{opt}). Aus $\mathcal{A}(X^{opt}) = \bar{b}$ und $X^{opt} \succeq 0$ folgt mit $C \bullet X^{opt} = \min!$, dass

$$X^{opt} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

die eindeutige Lösung von (P) ist. Aus $X^{opt}S^{opt} = 0$ und $S^{opt} = C - \mathcal{A}^*(y^{opt})$ erhalten wir

$$S^{opt} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

als eindeutige Lösung des Programmes (D). Damit ist Voraussetzung 3 erfüllt. Für dieses Beispiel gilt

$$\mathbf{L} = \left\{ \left(\begin{pmatrix} 0 & 0 & 0 \\ 0 & x_1 & x_2 \\ 0 & x_2 & x_3 \end{pmatrix}, \begin{pmatrix} s_1 & s_2 & s_3 \\ s_2 & 0 & 0 \\ s_3 & 0 & 0 \end{pmatrix} \right) \;\middle|\; x_1 + x_3 + s_1 = 0 \right\}$$

und $\mathbf{A} = \mathbf{L} + (X^{opt}, S^{opt})$. *Betrachten wir nun das Paar*

$$X_\varepsilon = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{pmatrix}, \; S_\varepsilon = \begin{pmatrix} -2\varepsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

für ein beliebiges $\varepsilon > 0$. *Es gilt* $(X_\varepsilon, S_\varepsilon) \in \mathbf{A}$. *Da* X_ε *und* S_ε *invertierbar sind, existiert nach Lemma 10 die zweite Ableitung* $\nabla^2\tilde{\phi}((X_\varepsilon, S_\varepsilon))$. *Für*

$$H := (\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}) \in \mathbf{L}$$

gilt

$$(\tilde{\phi} + \hat{f})((X_\varepsilon, S_\varepsilon) + \delta H) \equiv 2\varepsilon^2 \quad \forall \delta \in [-\varepsilon, \varepsilon].$$

Für jedes $\varepsilon > 0$ *folgt somit* $\nabla^2(\tilde{\phi} + \hat{f})((X_\varepsilon, S_\varepsilon))[H, H] = 0$.
Nach Korollar 3 ist die 2-Norm von $\nabla^2\tilde{\phi}((X_\varepsilon, S_\varepsilon))$ *beschränkt und da* $\hat{f}$ *beliebig oft differenzierbar ist, ist die 2-Norm von* $\nabla^2(\tilde{\phi}+\hat{f})((X_\varepsilon, S_\varepsilon))$ *ebenfalls beschränkt. Für eine Folge* $\varepsilon_k \searrow 0$ *besitzt die Folge* $\nabla^2(\tilde{\phi} + \hat{f})((X_{\varepsilon_k}, S_{\varepsilon_k}))$ *somit mindestens einen Häufungspunkt* Θ.
Es folgt dann $\Theta[H, H] = 0$ *und außerdem gilt* $\Theta \in \partial^2(\tilde{\phi} + \hat{f})((X^{opt}, S^{opt}))$.

Nach Satz 8 gilt $\partial^2(\tilde{\phi} + \hat{f})((X^{opt}, S^{opt})) \succeq 0$ – *also insbesondere* $\Theta \succeq 0$. *Damit ist* 0 *ein Eigenwert von* Θ *und* Θ *somit nicht invertierbar.*

Somit kann die verallgemeinerte Hessematrix der in [9] betrachteten Funktion nicht invertierbare Elemente enthalten – selbst wenn Voraussetzung 3 erfüllt ist. In [9] wurde aber auch die Verwendung der Funktion

$$\begin{array}{rccl} f: & \mathcal{S}^n \times \mathcal{S}^n & \to & I\!R, \\ & (X, S) & \mapsto & \|XS\|_F^2, \end{array}$$

als Regularisierung vorgeschlagen. Da $f(X, S) = \frac{1}{4}(\bar{f}(X, S) + \|XS + SX\|_F^2)$ wurde gefolgert, dass f mindestens genauso gute Eigenschaften wie $\bar{f}$ besitzt. Wir werden diese Funktion im Folgenden genauer betrachten:

Sei $\tilde{f} := f_{|\mathbf{A}}$. Eine Optimallösung (X^{opt}, S^{opt}) ist Minimalstelle der auf $\mathbf{A}$ beliebig oft differenzierbaren Funktion $\tilde{f}$. Es gilt also $\nabla^2\tilde{f}(X^{opt}, S^{opt}) \succeq 0$. Analog zu Lemma 13 und Satz 8 zeigt man:

Satz 10. *Die Funktion $\nabla(\tilde{\phi}+\tilde{f})$ ist lokal Lipschiz-stetig, stark semiglatt und fast überall differenzierbar auf* $\mathbf{A}$. *Es gilt weiterhin* $\partial^2(\tilde{\phi}+\tilde{f})(X^{opt}, S^{opt}) \succeq 0$.

Der Gradient der Funktion f ist von der Form

$$\nabla f(X,S) = \begin{pmatrix} S^2X + XS^2 \\ X^2S + SX^2 \end{pmatrix}, \tag{2.8}$$

so dass wir $\nabla\tilde{f}(X,S) = \Pi_{\mathbf{L}}(\nabla f(X,S))$ für $(X,S) \in \mathbf{A}$ erhalten.

Wir werden als Nächstes für die Funktion $\tilde{f}$ eine wichtige Aussage zeigen. Dazu nutzen wir den folgenden Satz, der für eine Verallgemeinerung der Funktion $\tilde{f}$ gilt:

Sei $E_{\bar{n}} := \mathcal{S}^{n_1} \times \cdots \times \mathcal{S}^{n_p}$ für ein $p \in \mathbb{N}$ und $\bar{n} := (n_1, \ldots, n_p) \in \mathbb{N}^p$. Das auf $E_{\bar{n}}$ gegebene Skalarprodukt sei definiert durch

$$\langle (A_1, \ldots, A_p), (B_1, \ldots, B_p) \rangle_{E_{\bar{n}}} := \sum_{i=1}^{p} A_i \bullet B_i.$$

Die zugehörige Norm ist dann gegeben durch

$$\|(A_1, \ldots, A_p)\|_{F_{\bar{n}}} = \sqrt{\sum_{i=1}^{p} \|A_i\|_F^2}.$$

Sei $\mathcal{K} := \mathcal{S}_+^{n_1} \times \cdots \times \mathcal{S}_+^{n_p}$. Dann gilt wie im Fall $p = 1$ auch hier $\mathcal{K}^D = \mathcal{K}$. Wir betrachten nun die in Abschnitt 1.1 definierten Programme (P) und (D) und die zugehörigen Räume $\mathbf{A}$, $\mathbf{L}$ und $\mathbf{K}$. Es gilt dann

$$\tilde{\phi}(X_1, \ldots X_p, S_1, \ldots, S_p) = \frac{1}{2} \sum_{i=1}^{p} \|X_i - \Pi_{\mathcal{S}_+^{n_i}}(X_i)\|_F^2 + \|S_i - \Pi_{\mathcal{S}_+^{n_i}}(S_i)\|_F^2.$$

Definieren wir die Verallgemeinerungen von $\tilde{f}$ durch

$$\tilde{f}_{\bar{n}}(X_1, \ldots X_p, S_1, \ldots, S_p) := \sum_{i=1}^{p} \tilde{f}(X_i, S_i)$$

für $(X_1, \ldots X_p, S_1, \ldots, S_p) \in \mathbf{A}$, dann gelten für $\tilde{\phi}$ und $\tilde{f}_{\bar{n}}$ alle Aussagen, die bisher in Kapitel 2 zu $\tilde{\phi}$ und $\tilde{f}$ gemacht wurden, da sie auf semidefinite Blöcke verallgemeinert werden können.

Voraussetzung 3 hat dann die folgende Form:
Seien Programme (P) und (D) gegeben, die Voraussetzung 1 erfüllen. Die Optimallösung $Z_{\bar{n}}^{opt} := (X_{\bar{n}}^{opt}, S_{\bar{n}}^{opt}) := ((X_1^{opt}, \ldots, X_p^{opt}), (S_1^{opt}, \ldots, S_p^{opt}))$ sei eindeutig und strikt komplementär, d.h. es gelte $X_i^{opt} + S_i^{opt} \succ 0$ für $1 \leq i \leq p$.

Satz 11. *Sei ein primal-duales Paar (P) und (D) gegeben. Weiterhin sei Voraussetzung 3 erfüllt.*
Sei $Z_{\bar{n}}^{opt}$ die zugehörige eindeutige und strikt komplementäre Optimallösung. Es gilt dann

$$\nabla^2 \tilde{f}_{\bar{n}}(Z_{\bar{n}}^{opt}) \succ 0.$$

Beweis. Da $((X_1^{opt}, \ldots, X_p^{opt}), (S_1^{opt}, \ldots, S_p^{opt})) \in \mathbf{A}$ komplementär ist und

$$\langle (X_1^{opt}, \ldots, X_p^{opt}), (S_1^{opt}, \ldots, S_p^{opt}) \rangle = 0 \Leftrightarrow X_i^{opt} \bullet S_i^{opt} = 0,\ 1 \leq i \leq p,$$

gilt, existieren orthogonale Matrizen $U_1, \ldots, U_p$, so dass

$$U_i^T X_i^{opt} U_i = \tilde{\Lambda}_i = \begin{pmatrix} \Lambda_i & 0 \\ 0 & 0 \end{pmatrix},\ U_i^T S_i^{opt} U_i = \tilde{\Sigma}_i = \begin{pmatrix} 0 & 0 \\ 0 & \Sigma_i \end{pmatrix},$$

mit positiv definiten Diagonalmatrizen Λ_i und Σ_i für $1 \leq i \leq p$ erfüllt ist, sofern der Fall $\tilde{\Lambda}_i = 0$ bzw. $\tilde{\Sigma}_i = 0$ nicht eintritt. Auf diesen gehen wir weiter unten ein. Aus der strikten Komplementarität folgt $\tilde{\Lambda}_i + \tilde{\Sigma}_i \succ 0$.

Sei $\Delta Z_{\bar{n}} := (\Delta Z_{\bar{n}}^X, \Delta Z_{\bar{n}}^S) := ((\Delta X_1, \ldots, \Delta X_p), (\Delta S_1, \ldots, \Delta S_p)) \in \mathbf{L}$ mit $\|\Delta Z_{\bar{n}}^X\|_{F_{\bar{n}}}^2 + \|\Delta Z_p^S\|_{F_{\bar{n}}}^2 = 1$ gegeben. Wir setzen

$$U_i^T \Delta X_i U_i =: \Delta \tilde{X}_i = \begin{pmatrix} \Delta \tilde{X}_{11}^{(i)} & \Delta \tilde{X}_{12}^{(i)} \\ \Delta \tilde{X}_{12}^{(i)T} & \Delta \tilde{X}_{22}^{(i)} \end{pmatrix},\ U_i^T \Delta S_i U_i =: \Delta \tilde{S}_i = \begin{pmatrix} \Delta \tilde{S}_{11}^{(i)} & \Delta \tilde{S}_{12}^{(i)} \\ \Delta \tilde{S}_{12}^{(i)T} & \Delta \tilde{S}_{22}^{(i)} \end{pmatrix}$$

für $1 \leq i \leq p$.
Es folgt

$$\begin{aligned}
& \tilde{f}_{\bar{n}}(Z_{\bar{n}}^{opt} + t\Delta Z_{\bar{n}}) \\
= & \sum_{i=1}^{p} \|(X_i^{opt} + t\Delta X_i)(S_i^{opt} + t\Delta S_i)\|_F^2 \\
= & \sum_{i=1}^{p} \|(\tilde{\Lambda}_i + t\Delta \tilde{X}_i)(\tilde{\Sigma}_i + t\Delta \tilde{S}_i)\|_F^2 \\
= & \sum_{i=1}^{p} \|t\tilde{\Lambda}_i \Delta \tilde{S}_i + t\Delta \tilde{X}_i \tilde{\Sigma}_i + t^2 \Delta \tilde{X}_i \Delta \tilde{S}_i\|_F^2 \qquad (2.9) \\
= & \sum_{i=1}^{p} \left\| t \begin{pmatrix} \Lambda_i \Delta \tilde{S}_{11}^{(i)} & \Lambda_i \Delta \tilde{S}_{12}^{(i)} \\ 0 & 0 \end{pmatrix} + t \begin{pmatrix} 0 & \Delta \tilde{X}_{12}^{(i)} \Sigma_i \\ 0 & \Delta \tilde{X}_{22}^{(i)} \Sigma_i \end{pmatrix} + t^2 \Delta \tilde{X}_i \Delta \tilde{S}_i \right\|_F^2 \\
= & \sum_{i=1}^{p} \left\| t \begin{pmatrix} \Lambda \Delta \tilde{S}_{11}^{(i)} & \Lambda_i \Delta \tilde{S}_{12}^{(i)} + \Delta \tilde{X}_{12}^{(i)} \Sigma_i \\ 0 & \Delta \tilde{X}_{22}^{(i)} \Sigma_i \end{pmatrix} \right\|_F^2 + \mathcal{O}(t^3) \\
= & t^2 \sum_{i=1}^{p} (\|\Lambda_i \Delta \tilde{S}_{11}^{(i)}\|_F^2 + \|\Lambda_i \Delta \tilde{S}_{12}^{(i)} + \Delta \tilde{X}_{12}^{(i)} \Sigma_i\|_F^2 + \|\Delta \tilde{X}_{22}^{(i)} \Sigma_i\|_F^2) + \mathcal{O}(t^3).
\end{aligned}$$

Setze

$$\Omega(\Delta Z_{\bar{n}}) := \sum_{i=1}^{p} \|\Lambda_i \Delta\tilde{S}_{11}^{(i)}\|_F^2 + \|\Lambda_i \Delta\tilde{S}_{12}^{(i)} + \Delta\tilde{X}_{12}^{(i)}\Sigma_i\|_F^2 + \|\Delta\tilde{X}_{22}^{(i)}\Sigma_i\|_F^2.$$

Um für $\tilde{f}_{\bar{n}}$ die Wachstumsbedingung 2. Ordnung nachzuweisen (welche nach Lemma 14 die positive Definitheit der Hessematrix implzieren würde) zeigen wir, dass $\Omega(\Delta Z_{\bar{n}}) > 0$ für alle $\Delta Z_{\bar{n}} \in \mathbf{L}$ mit $\|\Delta Z_{\bar{n}}^X\|_{F_{\bar{n}}}^2 + \|\Delta Z_{\bar{n}}^S\|_{F_{\bar{n}}}^2 = 1$ erfüllt ist:
Wir gehen zunächst auf den Fall ein, dass ein $i \in \{1, \ldots, p\}$ existiert, so dass $\tilde{\Lambda}_i = 0$ gilt. Der entsprechende Summand in (2.9) ist dann von der Form

$$t^2 \|\Delta\tilde{X}_i \tilde{\Sigma}_i\|_F^2 + \mathcal{O}(t^3).$$

Gilt $\Delta\tilde{X}_i \neq 0$, dann folgt wegen $\tilde{\Sigma}_i \succ 0$ gerade $\|\Delta\tilde{X}_i\tilde{\Sigma}_i\|_F^2 > 0$ und damit $\Omega(\Delta Z_{\bar{n}}) > 0$. Gilt dagegen $\Delta\tilde{X}_i = 0$, dann kann wegen $\Delta Z_{\bar{n}} \in \mathbf{L}$ und $\|\Delta Z_{\bar{n}}^X\|_{F_{\bar{n}}}^2 + \|\Delta Z_{\bar{n}}^S\|_{F_{\bar{n}}}^2 = 1$ der Fall $\Delta\tilde{X}_j = \Delta\tilde{S}_j = 0 \ \forall j \neq i$ ($\Rightarrow \Delta\tilde{S}_i \neq 0$) nicht eintreten, da sonst die Optimallösung nicht eindeutig wäre. Daher kann man einen solchen Index i in den folgenden Überlegungen ignorieren. Genau so verhält es sich, falls ein $i \in \{1, \ldots, p\}$ mit $\tilde{\Sigma}_i = 0$ existiert.
Falls ein $i \in \{1, \ldots, p\}$ existiert, so dass $\Delta\tilde{X}_{22}^{(i)} \neq 0$ oder $\Delta\tilde{S}_{11}^{(i)} \neq 0$ gilt, dann ist wegen $\Lambda_i \succ 0, \Sigma_i \succ 0$ nichts zu zeigen. Wir nehmen daher an, dass $\Delta\tilde{X}_{22}^{(i)} = 0$ und $\Delta\tilde{S}_{11}^{(i)} = 0$ für $1 \leq i \leq p$ gilt.
Falls ein $i \in \{1, \ldots, p\}$ existiert, so dass $\Delta\tilde{X}_{11}^{(i)} \neq 0$ gilt, dann existiert ein $j \in \{1, \ldots, p\}$, so dass $\Delta\tilde{X}_{12}^{(j)} \neq 0$ erfüllt ist; andernfalls wäre die Eindeutigkeit der Optimallösung $(X_1^{opt}, \ldots, X_p^{opt})$ für das primale Programm verletzt.
Falls ein $i \in \{1, \ldots, p\}$ existiert, so dass $\Delta\tilde{S}_{22}^{(i)} \neq 0$ gilt, dann existiert auch ein $j \in \{1, \ldots, p\}$, so dass $\Delta\tilde{S}_{12}^{(j)} \neq 0$ gilt; sonst wäre die Optimallösung $(S_1^{opt}, \ldots, S_p^{opt})$ für das duale Programm nicht eindeutig.
Gilt $\Delta\tilde{X}_{11}^{(i)} = 0$ und $\Delta\tilde{S}_{22}^{(i)} = 0$ für $1 \leq i \leq p$, dann folgt wegen $\|\Delta Z_{\bar{n}}^X\|_{F_{\bar{n}}}^2 + \|\Delta Z_{\bar{n}}^S\|_{F_{\bar{n}}}^2 = 1$ die Existenz eines $j \in \{1, \ldots, p\}$, so dass $\Delta\tilde{X}_{12}^{(j)} \neq 0$ oder $\Delta\tilde{S}_{12}^{(j)} \neq 0$ erfüllt ist.
In jedem Fall folgt somit, dass entweder $\Delta\tilde{X}_{12}^{(j)} \neq 0$ oder $\Delta\tilde{S}_{12}^{(j)} \neq 0$ für mindestens ein $j \in \{1, \ldots, p\}$ gilt.
Wir nehmen nun $\Lambda_i \Delta\tilde{S}_{12}^{(i)} + \Delta\tilde{X}_{12}^{(i)}\Sigma_i = 0$ for $1 \leq i \leq p$ an. Dann folgt $\Delta\tilde{S}_{12}^{(i)} = -\Lambda_i^{-1}\Delta\tilde{X}_{12}^{(i)}\Sigma_i$ für $1 \leq i \leq p$ und somit $\Delta\tilde{X}_{12}^{(j)} \neq 0$ und $\Delta\tilde{S}_{12}^{(j)} \neq 0$ für mindestens ein $j \in \{1, \ldots, p\}$.

Aus $(\Delta Z_{\bar{n}}^X, \Delta Z_{\bar{n}}^S) \in \mathbf{L} \subseteq \mathcal{L} \times \mathcal{L}^\perp$ folgt $\langle \Delta Z_{\bar{n}}^X, \Delta Z_{\bar{n}}^S \rangle = 0$. Wir erhalten damit

$$\begin{aligned} 0 &= \textstyle\sum_{i=1}^p \begin{pmatrix} \Delta\tilde{X}_{11}^{(i)} & \Delta\tilde{X}_{12}^{(i)} \\ \Delta\tilde{X}_{12}^{(i)T} & 0 \end{pmatrix} \bullet \begin{pmatrix} 0 & \Delta\tilde{S}_{12}^{(i)} \\ \Delta\tilde{S}_{12}^{(i)T} & \Delta\tilde{S}_{22}^{(i)} \end{pmatrix} \\ &= 2\textstyle\sum_{i=1}^p \Delta\tilde{X}_{12}^{(i)} \bullet \Delta\tilde{S}_{12}^{(i)} = -2\sum_{i=1}^p \Delta\tilde{X}_{12}^{(i)} \bullet \Lambda_i^{-1}\Delta\tilde{X}_{12}^{(i)}\Sigma_i \\ &= -2\textstyle\sum_{i=1}^p \|\Sigma_i^{\frac{1}{2}}\Delta\tilde{X}_{12}^{(i)T}\Lambda_i^{-\frac{1}{2}}\|_F^2 < 0 \end{aligned}$$

und damit einen Widerspruch. Damit war die Annahme falsch. Insgesamt erhalten wir

$$\Omega(\Delta Z_{\bar{n}}) > 0 \ \forall \ (\Delta Z_{\bar{n}}^X, \Delta Z_{\bar{n}}^S) \in \mathbf{L}, \ \|\Delta Z_{\bar{n}}^X\|_{F_{\bar{n}}}^2 + \|\Delta Z_{\bar{n}}^S\|_{F_{\bar{n}}}^2 = 1.$$

Da Ω eine stetige Funktion ist, folgt weiterhin

$$\mu := \inf\{\Omega(\Delta Z_{\bar{n}}) \ \mid \ (\Delta Z_{\bar{n}}^X, \Delta Z_{\bar{n}}^S) \in \mathbf{L}, \ \|\Delta Z_{\bar{n}}^X\|_{F_{\bar{n}}}^2 + \|\Delta Z_{\bar{n}}^S\|_{F_{\bar{n}}}^2 = 1\} > 0.$$

Mit 2.9 folgt

$$\tilde{f}_{\bar{n}}(Z_{\bar{n}}^{opt} + t\Delta Z_{\bar{n}}) \geq t^2\mu + \mathcal{O}(t^3),$$

womit $\tilde{f}_{\bar{n}}$ die WG2 erfüllt. Mit Lemma 14 erhalten wir daher $\nabla^2 \tilde{f}(Z_{\bar{n}}^{opt}) \succ 0$. $\square$

Wir schränken uns ab hier wieder auf den Fall $p = 1$ ein:
Satz 11, Satz 7 und (2.4) (für $\tilde{f}$ anstatt für $\hat{f}$) implizieren die folgende Verschärfung von Satz 10:

$$\partial^2(\tilde{\phi} + \tilde{f})(X^{opt}, S^{opt}) \succ 0. \tag{2.10}$$

Damit sind für die Funktion $\tilde{\phi}+\tilde{f}$ die Voraussetzungen von Satz 4 erfüllt. Es folgt: Nutzt man unter Voraussetzung 3 Algorithmus 2 zur Minimierung von $\tilde{\phi}+\tilde{f}$, dann konvergiert dieser lokal quadratisch gegen die Optimallösung Z^{opt}. Das verallgemeinerte Newton-Verfahren bzw. ein verallgemeinertes Quasi-Newton-Verfahren lässt sich in Algorithmus 1 zur Suchschrittbestimmung einbinden. Um (2.10) zu nutzen schlagen wir die folgende aus [9] abgeleitete Vorgehensweise vor:

1. Minimiere $\tilde{\phi}$ um sich der Optimallösung ausreichend anzunähern.
2. Minimiere danach $\tilde{\phi} + \tilde{f}$ um die Optimallösung zu bestimmen.

2.5 Hochdimensionale Programme

Die Resultate aus Abschnitt 2.4 lassen sich verwenden um die Optimallösung eines gegebenen primal-dualen Paares (P) und (D) zu approximieren. Das verallgemeinerte Newton-Verfahren zur Minimierung von $\tilde{\phi} + \tilde{f}$ wird unter Voraussetzung 3 lokal quadratisch konvergieren. Für hochdimensionale Probleme ist es jedoch sehr teuer oder gar nicht mehr verwendbar, da die Berechnung eines Newton-Schrittes einen Speicherplatz in $\mathcal{O}(n^4)$ (Speicherung der Hessematrix) und schlimmstenfalls einen Rechenaufwand von $\mathcal{O}(n^6)$ (Bestimmung der Suchrichtung durch Lösung eines Gleichungssystems) erfordert. Günstiger sind (iterative) Methoden erster Ordnung wie z.B. das Verfahren des steilsten Abstiegs. Um dieses und auch andere für den $\mathbb{R}^d$ gegebene Verfahren auf dem Raum $\mathcal{S}^n$ (oder auf einem affinen Teilraum davon) zu nutzen, wird ein isometrischer Isomorphismus von $\mathcal{S}^n$ nach

$\mathbb{R}^d$ mit $d = n(n+1)/2$ gewählt. Es muss eine Isometrie sein, da bei den meisten für den $\mathbb{R}^d$ entwickelten Verfahren der Winkel zwischen zwei Vektoren von Bedeutung ist. Wir definieren:

$$\begin{array}{rccl} \mathsf{svec}: & \mathcal{S}^n & \to & \mathbb{R}^{\frac{n(n+1)}{2}}, \\ & X & \mapsto & ((\delta_{ij}(1-\sqrt{2})+\sqrt{2})X_{ij})_{\substack{i=1\ldots n,\\ j=i\ldots n}}, \end{array}$$

δ_{ij} repräsentiert hier das Kronecker-Delta.

Diese Abbildung ist ein Isomorphismus. Die Umkehrabbildung wird mit smat bezeichnet. svec ist eine Isometrie:
Seien $A,\ B \in \mathcal{S}^n$ und $a,\ b \in \mathbb{R}^{\frac{n(n+1)}{2}}$ mit $A = \mathsf{smat}(a)$, $B = \mathsf{smat}(b) \Leftrightarrow a = \mathsf{svec}(A)$, $b = \mathsf{svec}(B)$. Es gilt dann

$$A \bullet B = a^T b \text{ und somit } \|A\|_F = \|a\|_2.$$

Methoden erster Ordnung konvergieren jedoch oftmals sehr langsam, so dass sie nicht zur Berechnung von Lösungen mit sehr hoher Genauigkeit geeignet sind. Ein anderes Verfahren, welches ebenfalls verwendet werden kann, ist der cg-Algorithmus von Fletcher-Reeves und seine Varianten wie z.B. die von Polak-Ribiere (siehe [16]). Bisher ist der cg-Algorithmus jedoch nur für quadratische strikt konvexe Funktionen genau verstanden worden. Für allgemeine Funktionen kann die Konvergenz sehr langsam sein.
Eine weitere Möglichkeit ist die Verwendung von Quasi-Newton-Verfahren, welche die Hessematrix bzw. deren Inverse approximieren. Diese haben i.Allg. jedoch einen höheren Speicherbedarf und benötigen mehr Rechenzeit – auch wenn der Bedarf niedriger und der Aufwand geringer als beim Newton-Verfahren ausfällt. Besonders herausstellen wollen wir hierbei das BFGS-Verfahren. Wir stellen es im Folgenden kurz vor. Eine ausführlichere Betrachtung findet sich z.B. in [10] oder [16].

Sei eine stetig differenzierbare Funktion $\eta : \mathbb{R}^d \to \mathbb{R}$ gegeben, die minimiert werden soll. Weiterhin sei ein Punkt $x \in \mathbb{R}^d$ und eine positiv definite Matrix H, die als Näherung für $\nabla^2\eta(x)^{-1}$ (falls existent!) dienen soll, gegeben. Für eine gegebene Suchrichtung $s \in \mathbb{R}^d$ und $y := \nabla\eta(x+s) - \nabla\eta(x)$ hat das BFGS-Update die folgende Form:

$$H_+ := H + \frac{(s^T y + y^T H y) s s^T}{(s^T y)^2} - \frac{H y s^T + s y^T H}{s^T y}.$$

Falls $s^T y > 0$ gilt, dann ist H_+ wieder positiv definit. Diese Matrix soll als neue Approximation für $\nabla^2\eta(x+s)^{-1}$ dienen und es gilt die Quasi-Newton-Bedingung $H_+ y = s$. Die neue Suchrichtung ist dann von der Form $s_+ := -H_+\nabla\eta(x)$ und die neue Iterierte ist gegeben durch $x_+ := x + \rho s_+$, wobei ρ eine sinnvolle Schrittweite

ist, die i.Allg. mit Hilfe einer linesearch ermittelt wird. Wenn im BFGS-Verfahren ein (positives) Vielfaches der Identität als Startmatrix gewählt wird, dann ist das Verfahren affin invariant.
Bei Verwendung des Verfahrens innerhalb von Algorithmus 1 liegt der Speicherbedarf in jeder Iteration in $\mathcal{O}(n^4)$, der Rechenaufwand liegt ebenfalls in $\mathcal{O}(n^4)$. Dies ist für hochdimensionale Probleme aber noch immer sehr hoch. Für solche Probleme nutzt man daher zwei aufeinander aufbauende Modifikationen des BFGS-Verfahrens:
Die erste Modifikation ist die Verwendung einer "limited memory"-Variante von BFGS (im Folgenden: L-BFGS). So werden in jeder Iteration höchstens die letzten k Iterierten beim Matrixupdate berücksichtigt. Alle vorherigen Informationen werden verworfen. Die zweite Modifikation ist die Verwendung einer alternativen Generierung der aus dem BFGS-Update resultierenden Suchrichtung – hierfür ist die Verwendung einer Vielfachheit der Identität als Startmatrix im BFGS-Update notwendig. Eine Untersuchung dieser Modifikationen findet sich in [16]. Wir geben hier den zugehörigen Algorithmus an:

Algorithmus 3 (L-BFGS-Update)**.** *Sei x_j die aktuelle Iterierte und $\bar{I} := \delta I$ für ein $\delta > 0$.*
1) Setze $q := \nabla\eta(x_j)$.
2) Für $i = j-1, \ldots, j-k$ setze
$\alpha_i := \frac{s_i^T q}{s_i^T y_i}$,
$q := q - \alpha_i y_i$.
3) Setze $r := \bar{I}q$.
4) Für $i = j-k, \ldots, j-1$ setze
$\beta := \frac{y_i^T r}{s_i^T y_i}$,
$r := r + s_i(\alpha_i - \beta)$.
5) Setze $s_{j+1} := -r$ und $x_{j+1} := x_j + \rho s_{j+1}$ für eine durch eine linesearch ermittelte Schrittweite $\rho > 0$.

Gilt $k = j-1$, dann entspricht Algorithmus 3 dem BFGS-Update mit $H_0 := \bar{I}$. In diesem Fall gilt also $s_{j+1} = -H_{j+1}\nabla\eta(x_j)$, wobei H_{j+1} durch die Formel $H_{l+1} = (H_l)_+$ für $0 \leq l \leq j$ erzeugt wird.
Der Speicherbedarf liegt bei Verwendung dieser Modifikationen in $\mathcal{O}(n^2)$, der Rechenaufwand liegt dagegen in $\mathcal{O}(kn^2)$. Wird k sehr viel kleiner als n und im gesamten Verfahren beschränkt, dann kann es bei der Aufwandsabschätzung vernachlässigt werden.

Wir gehen wieder auf die Minimierung der Funktion $\tilde{\phi}$ und den damit verbundenen Aufwand ein. Wird Algorithmus 1 mit L-BFGS zur Minimierung von $\tilde{\phi}$ verwendet, dann hängt der Rechenaufwand im Wesentlichen von drei Faktoren ab: Einmal von dem oben bschriebenden L-BFGS-Update und weiterhin von den Projektionen auf $\mathbf{L}$ bzw. auf $\mathbf{K}$ (siehe Abschnitt 1.2). Der für $\Pi_{\mathbf{L}}$ bedeutende

Unterraum $\mathcal{L}$ werde durch den Nullraum eines Operators $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ mit $m \leq n(n+1)/2$ repräsentiert. Dieser wiederum lässt sich mit Hilfe einer Matrix $A \in \mathbb{R}^{m \times \frac{n(n+1)}{2}}$ (mittels svec) darstellen. Ist A voll besetzt, dann liegt der Speicherbedarf zum Festhalten der Informationen in $\mathcal{O}(mn^2)$ und ist damit für hochdimensionale Probleme sehr groß. Der Wesentliche Teil der Rechenkosten liegt in der Abbildung $\Pi_{\mathcal{L}}(X) = X - \mathcal{A}^*((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(X))$ und einer entsprechenden Abbildung für die die duale Komponente S, wobei vorausgesetzt sei, dass $(\mathcal{A}\mathcal{A}^*)^{-1}$ existiert. Unter dieser Voraussetzung erfordert die einmalige Bestimmung eines Cholesky Faktors L von $\mathcal{A}\mathcal{A}^*$ $\mathcal{O}(m^3)$ Schritte. Die Berechnungskosten von $\Pi_{\mathcal{L}}(X)$ liegen dann in $\mathcal{O}(mn^2)$. Für hohe Dimensionen ist der Aufwand damit zu groß.

Ist A jedoch dünn besetzt und wird entsprechend implementiert, dann ist der Speicherbedarf niedriger. Er entspricht im Wesentlichen der Anzahl der Nicht-Null-Einträge von A (und damit der von $\mathcal{A}$). Weiterhin sind dann oftmals auch AA^T und der Cholesky Faktor L von AA^T dünn besetzt. In einem solchen Fall ist die Anzahl der Nicht-Null-Einträge von AA^T und von L für den Rechenaufwand zur Bestimmung von $\Pi_{\mathcal{L}}$ ausschlaggebend. Ist diese Anzahl relativ klein, dann ist die Projektion auf **L** bzw. auf **A** nicht so teuer wie die Projektion auf **K**, deren Berechnung in $\mathcal{O}(n^3)$ liegt - siehe Abschnitt 2.3. Ist A dünn besetzt und k aus dem L-BFGS-Update sehr klein, dann liegt der Speicherbedarf des gesamten Verfahrens typischerweise in $\mathcal{O}(n^2)$.

2.6 Numerische Ergebnisse

Wir gehen nun auf die Implementierung der APD-Methode ein: Anstatt bei der Bestimmung der Optimallösung von einem gegebenen primal-dualen Paar (P) und (D) zuerst $\tilde{\phi}$ und danach $\tilde{\phi} + \tilde{f}$ zu minimieren, kann stattdessen von Beginn an die Funktion $\tilde{\phi} + \alpha\tilde{f}$ mit einem $\alpha \geq 0$ verwendet werden. Dieser zusätzliche Parameter spiegelt die Gewichtung von $\tilde{f}$ wieder. Er kann im Algorithmus anhand einiger Kriterien erhöht oder verkleinert werden (sollte der Algorithmus z.B. eine Suchrichtung generieren, die für $\tilde{\phi}$ alleine keine Abstiegsrichtung ist, dann wird $\tilde{f}$ zu stark gewichtet und α sollte verkleinert werden). Bei konstantem α ist es möglich, dass der Algorithmus gegen ein lokales aber nicht globales Minimum der Funktion $\tilde{\phi} + \alpha\tilde{f}$ konvergiert, siehe [8].

Wie in Abschnitt 2.5 erwähnt wurde, ist für Probleme hoher Dimension mit dünn besetzter "Datenmatrix" die Projektion auf **K** oftmals die teuerste Operation in Algorithmus 1, falls L-BFGS mit relativ kleinem "Speicher" zur Erzeugung der Suchschritte verwendet wird. Abseits der linesearch kommt man mit einer einzigen Projektion auf **K** pro Iteration aus. Um auch innerhalb der linesearch die Anzahl der Projektionen gering zu halten kann $\tilde{\phi} + \alpha\tilde{f}$ entlang der Suchrichtung interpoliert werden.

Da Methoden erster Ordnung i.Allg. nicht skalierungsinvariant sind, werden

Name	n	m	$nnz(\mathcal{A})$ (%)
P1	400	30000	179904 (0.004)
P2	700	50000	298404 (0.001)
P3	1000	100000	596964 (0.001)

Tabelle 2.1: Problemdaten

die Eingabedaten folgendermaßen äquivalent umformuliert:

Zuerts einmal werden B auf $\mathcal{L}^{\perp}$ und C auf $\mathcal{L}$ projiziert. Dadurch lassen sich die Projektionen auf $\mathbf{L}$ und $\mathbf{A}$ leicht berechnen, wie in Abschnitt 1.2 ausführlich beschrieben wurde. Um das primale und das duale Problem gleichwertig zu behandeln, werden B und C durch $B/\|B\|_F$ und $C/\|C\|_F$ ersetzt. Ist ein Cholesky Faktor von AA^T gegeben, dann sind diese (einmaligen) Operationen nicht teuer.

Abschließend geben wir einige numerische Ergebnisse wieder, die mit dem APD-Verfahren bei Verwendung des steilsten Abstiegs, des cg-Verfahrens bzw. des L-BFGS-Verfahrens zur Bestimmung der Suchrichtung erzielt wurden. Dazu betrachten wir drei Probleme unterschiedlicher Dimension, welche alle Voraussetzung 3 erfüllen. Sie sind außerdem so gewählt, dass die zugehörigen Restriktionen durch dünnbesetzte Matrizen dargestellt werden können. Die Eckdaten der Probleme werden in Tabelle 2.1 angegeben. Dabei steht n für die Dimesnion des Problems, m für die Anzahl der Nebenbedingungen und $nnz(\mathcal{A})$ (%) beschreibt die absolute und relative Anzahl (in %) der Nicht-Null-Einträge in der zu den Restriktionen gehörenden Datenmatrix.
Alle Beispiele wurden mit einer MATLAB® Implementierung auf einem TEST-PC mit acht Intel® Xeon® X3470/2.93GHz Prozessoren und 16 GB RAM berechnet. (Die Matlab Version verwendete jedoch lediglich einen der Kerne zur Bearbeitung der Probleme.)

In Tabelle 2.2 wird Problem 1 betrachtet. Es wurden mit jeder der drei oben beschriebenen APD-Varianten 500 Iterationen berechnet. Da die Iterierten (X, S) wegen der Definition des Verfahrens nur die Kegelbedingung verletzen und zu Beginn des Verfahrens die oben beschriebenen Skalierungen der Daten B und C vorgenommen wurden, wird der primal-duale Fehler durch

$$error := |\lambda_{min}(\tilde{X})| + |\lambda_{min}(\tilde{S})| \tag{2.11}$$

gemessen. $(\tilde{X}, \tilde{S})$ sind die analog zu B und C umskalierten Daten.

Weiterhin bezeichnet $\lambda_{min}(\tilde{X})$ den kleinsten Eigenwert von $\tilde{X}$ und $\lambda_{min}(\tilde{S})$ den kleinsten Eigenwert von $\tilde{S}$.
In der Spalte it von Tabelle 2.2 wird die Iterationsnummer angegeben. In der Spalte CPU_{sd} wird die Laufzeit in Sekunden festgehalten, die die *steilster Abstieg*-Variante bis zur Berechnung der aktuellen Iterierten bereits benötigt hat. Die

it	CPU_{sd}	$error_{sd}$	CPU_{cg}	$error_{cg}$	CPU_{L}	$error_{L}$
1	12.6	6.2081e-02	11.9	6.2081e-02	11.9	6.2081e-02
50	185.5	3.5702e-03	234.7	4.6188e-04	201.6	3.6719e-04
100	358.6	1.6987e-03	428.3	8.7546e-05	359.1	6.5000e-05
200	691.2	7.3494e-04	795.5	1.8208e-05	670.0	1.1264e-05
300	1014.8	4.6842e-04	1158.4	8.4163e-06	1014.1	4.0002e-06
400	1333.5	3.2217e-04	1518.9	4.9744e-06	1307.2	1.8719e-06
500	1646.8	2.4293e-04	1878.6	3.2307e-06	1602.0	9.8618e-07

Tabelle 2.2: P1 - Vergleich der APD-Varianten

it	CPU_{sd}	$error_{sd}$	CPU_{cg}	$error_{cg}$	CPU_{L}	$error_{L}$
1	13.4	4.6124e-02	14.6	4.6124e-02	12.0	4.6124e-02
50	376.6	3.7739e-03	456.9	8.9621e-04	395.7	8.1572e-04
100	749.1	2.5609e-03	901.4	2.5211e-04	759.0	2.2293e-04
200	1435.3	1.7259e-03	1767.4	5.7600e-05	1461.7	3.6915e-05
300	2118.1	1.2790e-03	2645.1	2.5186e-05	2161.2	1.0973e-05
400	2802.0	1.0001e-03	3491.9	1.2669e-05	2936.1	4.6692e-06
500	3480.9	8.0004e-04	4394.9	6.2997e-06	3632.0	2.5794e-06

Tabelle 2.3: P2 - Vergleich der APD-Varianten

it	CPU_{sd}	$error_{sd}$	CPU_{cg}	$error_{cg}$	CPU_{L}	$error_{L}$
1	31.5	3.9287e-02	35.0	3.9287e-02	32.0	3.9287e-02
50	963.4	3.1536e-03	1026.2	1.0435e-03	867.0	7.1474e-04
100	1932.1	2.2819e-03	2019.6	2.8756e-04	1676.3	2.1627e-04
200	3843.4	1.5557e-03	4305.2	6.2375e-05	3327.0	4.0876e-05
300	5371.9	1.1676e-03	6279.6	2.9729e-05	4860.0	1.2847e-05
400	6902.9	9.6534e-04	8439.2	1.7669e-05	6382.6	5.8994e-06
500	8417.5	8.3291e-04	10458.8	1.1420e-05	7906.1	3.2622e-06

Tabelle 2.4: P3 - Vergleich der APD-Varianten

Spalte $error_{sd}$ gibt den primal-dualen Fehler der vom *steilster Abstieg*-Verfahren berechneten Iterierten an. Entsprechend sind die Spalten CPU_{cg} und $error_{cg}$ für das *cg*-Verfahren und die Spalten CPU_L und $error_L$ für das *L-BFGS*-Verfahren definiert. Beim Letzteren wurde der Speicher auf 10 gesetzt. Bei der Berechnung der Suchrichtung werden also die letzten 10 Iterierten berücksichtigt.
Tabelle 2.3 ist analog zu Tabelle 2.2 für Problem 2 aufgebaut. Entsprechend bezieht sich Tabelle 2.4 auf Problem 3.

Wie erwartet ist die *steilster Abstieg*-Variante die schlechteste Wahl. Die *cg*-Variante funktionierte besser – musste jedoch für die linesearch mehr Zeit investieren als das *L-BFGS*-Verfahren, da die *cg*-Version die Krümmung der Funktion schlechter approximierte und daher mehr Funktionsauswertungen benötigt wurden. Diese sind der teuerste Teil des APD-Verfahrens, da sie die Berechnung der Projektion auf den Kegel $\mathbf{K}$ bzw. auf $\mathcal{S}_+^n$ erfordern. Ein weiterer Nachteil des *cg*-Verfahrens ist der erforderliche $Neustart$, sobald die berechneten Gradienten ihre Orthogonalitätseigenschaft verlieren. In den getesten Beispielen schneidet das *L-BFGS*-Verfahren durchgängig am besten ab. Es liefert als Verfahren erster Ordnung selbst für hochdimensionale Probleme Approximationen mit einem annehmbaren Fehler.

Kapitel 3

Verfahren zweiter Ordnung

Ist man für ein primal-duales Paar (P) und (D) hoher Dimensionen an einer sehr genauen Lösung interessiert, dann reichen die in Kapitel 2 vorgestellten Methoden zur Bestimmung der Suchrichtung i.Allg. nicht mehr aus. In diesem Kapitel werden wir zwei Erweiterungen der APD-Methode präsentieren, mit denen man ein besseres Konvergenzverhalten auf Kosten eines erhöhten Speicherbedarfs und Rechenaufwandes erhält.

3.1 Verallgemeinertes Newton-Verfahren

In Algorithmus 2 wurde das verallgemeinerte Newton-Verfahren bereits beschrieben. Es lässt sich natürlich mit Algorithmus 1 kombinieren, in dem es zur Suchrichtungsbestimmung verwendet wird und anschließend ggf. eine linesearch ausgeführt wird. Für die weitere Untersuchung dieser Methode setzen wir

$$\psi(X,S) := \tilde{\phi}(X,S) + \tilde{f}(X,S)$$

für $(X,S) \in \mathbf{A}$.
Wie in Abschnitt 2.5 bereits erklärt wurde, ist die tatsächliche Speicherung eines Elements $M \in \partial^2\psi(X,S)$, sowie die Bestimmung der Suchrichtung $W = -M^{-1}[\nabla\psi(X,S)]$ durch eine direkte Lösung des Systems

$$M[W] = -\nabla\psi(X,S) \tag{3.1}$$

für hochdimensionale Probleme zu "teuer". Die Suchrichtung kann jedoch iterativ approximiert werden. Im Folgenden gehen wir auf den damit verbundenen Speicherbedarf und Rechenaufwand ein:

Die Variante des cg-Verfahrens, welche wir zur Lösung des Gleichungssystems (3.1) nutzen wollen, benötigt die Richtungsableitungen von $\nabla\tilde{\phi}(X,S)$ und $\nabla\tilde{f}(X,S)$. Erstere haben wir in (2.2) und in Lemma 11 bereits angegeben. Wir

verwenden die Darstellung (2.8) um auch die Richtungsableitungen von $\nabla \tilde{f}(X,S)$ anzugeben. Sei dazu $(H_1, H_2) \in \mathbf{L}$ gegeben. Es folgt

$$\nabla^2 \tilde{f}(X,S)[(H_1,H_2)] = \lim_{t\to 0} \frac{1}{t}(\nabla \tilde{f}(X+tH_1, S+tH_2) - \nabla \tilde{f}(X,S)).$$

Wir betrachten zunächst die Teilfunktionen von ∇f genauer. (Es gilt $\nabla \tilde{f} = \Pi_{\mathbf{L}} \circ \nabla f$.) Wir setzen

$$\nabla f(X,S) = \begin{pmatrix} S^2X + XS^2 \\ X^2S + SX^2 \end{pmatrix} =: \begin{pmatrix} f_X(X,S) \\ f_S(X,S) \end{pmatrix}.$$

Es folgt:

$$\begin{aligned}
&f_X(X+tH_1, S+tH_2) - f_X(X,S) \\
&= (S+tH_2)^2(X+tH_1) + (X+tH_1)(S+tH_2)^2 - f_X(X,S) \\
&= (S^2 + t(SH_2 + H_2S) + t^2H_2^2)(X+tH_1) + (X+tH_1)(S+tH_2)^2 - f_X(X,S) \\
&= t(SH_2X + H_2SX + XSH_2 + XH_2S + S^2H_1 + H_1S^2) \\
&\quad + t^2(SH_2H_1 + H_2SH_1 + H_1SH_2 + H_1H_2S + H_2^2X + XH_2^2) \\
&\quad + t^3(H_2^2H_1 + H_1H_2^2).
\end{aligned}$$

Es folgt somit

$$\begin{aligned}
&\nabla f_X(X,S)[(H_1,H_2)] \\
&= \lim_{t\to 0} \tfrac{1}{t}(f_X(X+tH_1, S+tH_2) - f_X(X,S)) \\
&= SH_2X + H_2SX + XSH_2 + XH_2S + S^2H_1 + H_1S^2 \\
&= (SH_2 + H_2S)X + X(SH_2 + H_2S) + S^2H_1 + H_1S^2.
\end{aligned}$$

Analog zeigt man

$$\begin{aligned}
&\nabla f_S(X,S)[(H_1,H_2)] \\
&= \lim_{t\to 0} \tfrac{1}{t}(f_S(X+tH_1, S+tH_2) - f_S(X,S)) \\
&= XH_1S + H_1XS + SXH_1 + SH_1X + X^2H_2 + H_2X^2 \\
&= (XH_1 + H_1X)S + S(XH_1 + H_1X) + X^2H_2 + H_2X^2.
\end{aligned}$$

Insgesamt erhalten wir somit

$$\nabla^2 \tilde{f}(X,S)[(H_1,H_2)] = \Pi_{\mathbf{L}}(\nabla f_X(X,S)[(H_1,H_2)], \nabla f_S(X,S)[(H_1,H_2)]).$$

Sind Programme (P) und (D) gegeben, für die Voraussetzung 3 erfüllt ist, dann gilt $\partial^2\psi(X^{opt}, S^{opt}) \succ 0$. Nach Satz 4 existiert eine Umgebung $\mathcal{B}$ von (X^{opt}, S^{opt}), in der das verallgemeinerte Newton-Verfahren quadratisch konvergiert und weiterhin $\partial^2\psi(X,S) \succ 0$ für alle $(X,S) \in \mathcal{B}$ gilt. Letzteres folgt insbesondere aus den Sätzen 7 und 11, da $\nabla^2\tilde{f}$ auf $\mathbf{A}$ stetig ist. Ist $(X,S) \in \mathcal{B}$ gegeben, dann ist $\nabla\psi$ dort fast sicher differenzierbar. Im Falle der Differenzierbarkeit kann zur

Bestimmung der Newton-Richtung $\nabla^2\psi(X,S) \in \partial^2\psi(X,S)$ gewählt werden. Eine Auswertung der Form

$$\nabla^2\psi(X,S)[(H_1,H_2)] = \nabla^2\tilde{\phi}(X,S)[(H_1,H_2)] + \nabla^2\tilde{f}(X,S)[(H_1,H_2)]$$

erfordert dabei die folgenden Rechenoperationen: Der $\tilde{\phi}$-Anteil benötigt 8 Matrix-Matrix Multiplikationen (Aufwand: jeweils n^3) und weitere Operationen in $\mathcal{O}(n^2)$. Um den $\tilde{f}$-Anteil zu berechnen müssen 4 Matrix-Matrix Multiplikationen und einige weitere Rechnungen in $\mathcal{O}(n^2)$ ausgeführt werden. Innerhalb der gesamten Auswertung wird einmal auf $\mathbf{L}$ projiziert. Der Speicherbedarf liegt oftmals in $\mathcal{O}(n^2)$, sofern $\mathcal{L}$ mit Hilfe eines dünnbesetzten Operators beschrieben werden kann.

Um einen Newton-Schritt zu approximieren ohne die Hessematrix explizit aufzustellen, können verschiedene iterative Verfahren genutzt werden. Die Verwendung des cg-Verfahrens in der Umgebung $\mathcal{B}$ ist sinnvoll, da dort die (verallgemeinerte) Hessematrix positiv definit ist. Soll das verallgemeinerte Newton-Verfahren auch schon in einem Bereich erlaubt werden, in dem die positive Definitheit von $\partial^2\psi(X,S)$ nicht gesichert ist, dann wählen wir eine Variante des cg-Verfahrens, die auch in solchen Fällen wohldefiniert ist – der cg-Algorithmus von Steihaug:

Gegeben sei eine zweimal differenzierbare Funktion $\eta : \mathbb{R}^n \to \mathbb{R}$. Gesucht ist die Lösung des Problems

$$(TR) \qquad \min_{p:\|p\|_2 \le \Delta} \eta(x) + \nabla\eta(x)^T p + \frac{1}{2} p^T M p,$$

wobei $M = M^T$ eine Approximation von $\nabla^2\eta(x)$ sei und $\nabla\eta(x) \neq 0$ gelte.

Algorithmus 4 (cg-Steihaug)**.** *Sei $x \in \mathbb{R}^n$ gegeben. Wähle einen Genauigkeitsparameter $\varepsilon > 0$ und setze $g := \nabla\eta(x)$ und weiterhin $p_0 := 0$, $r_0 := g$, $d_0 := -r_0$.*
1) Falls $\|r_0\|_2 < \varepsilon$: Gebe $p := p_0$ aus.
2) Für $j = 0, 1, 2, \ldots$
Falls $d_j^T M d_j \le 0$
Bestimme τ, so dass $(p_j + \tau d_j)^T g < 0$ und $\|p_j + \tau d_j\|_2 = \Delta$ gilt.
Gebe $p := p_j + \tau d_j$ aus.
Setze $\alpha_j := \frac{r_j^T r_j}{d_j^T M d_j}$.
Setze $p_{j+1} := p_j + \alpha_j d_j$.
Falls $\|p_{j+1}\|_2 \ge \Delta$
Bestimme $\tau \ge 0$, so dass $\|p_j + \tau d_j\|_2 = \Delta$ gilt.
Gebe $p := p_j + \tau d_j$ aus.
Setze $r_{j+1} := r_j + \alpha_j M d_j$.
Falls $\|r_{j+1}\|_2 < \varepsilon\|r_0\|_2$
Gebe $p := p_{j+1}$ aus.

Setze $\beta_j := \frac{r_{j+1}^T r_{j+1}}{r_j^T r_j}$.
Setze $d_{j+1} := r_{j+1} + \beta_{j+1} d_j$.

Der Unterschied zum normalen cg-Algorithmus ist einmal, dass eine Art Trust-Region (Vertrauensbereich) für die zu ermittelnde Lösung vorgegeben werden kann, und weiterhin, dass M auch negative Eigenwerte beinhalten und singulär sein darf. Bricht das Verfahren wegen der Bedingung $\|r_{j+1}\|_2 < \varepsilon \|r_0\|_2$ ab, dann erfüllt die Lösung p die Bedingung $\|Mp + g\|_2 / \|g\|_2 < \varepsilon$, es gilt also $p \approx -M^{-1}g$, falls M invertierbar ist. Gilt dabei $M = \nabla^2 \eta(x)$, dann entspricht p in etwa dem Newton-Schritt. Eine ausführliche Beschreibung des Verfahrens kann in [16] gefunden werden.

In jeder Iteration von Algorithmus 4 muss einmal das Produkt Md_j berechnet werden – alle weiteren Operationen sind Vektor-Vektor Multiplikationen. Verwenden wir Algorithmus 4 zur Berechnung des (verallgemeinerten) Newton-Schrittes bei der Minimierung von ψ, dann kostet die dem Produkt Md_j entsprechende Auswertung 12 Matrix-Matrix Multipliaktionen und eine Projektion auf $\mathbf{L}$ (siehe oben). Dieser Aufwand muss in jeder Iteration von Algorithmus 4 betrieben werden. Der restliche Rechenaufwand und der Speicherbedarf liegen in $\mathcal{O}(n^2)$ und können daher vernachlässigt werden. Der Algorithmus lässt sich dabei wieder mit Hilfe des svec-Operators von $\mathbb{R}^{n(n+1)}$ auf den Raum $\mathcal{S}^n \times \mathcal{S}^n$ übertragen.

Versucht man von Beginn an, das verallgemeinerte Newton-Verfahren zur Minimierung von ψ zu verwenden, dann läuft man Gefahr lokale aber nicht globale Minimalstellen zu approximieren (siehe dazu [8]). Da das in Abschnitt 2.5 vorgestellte L-BFGS-Verfahren zu Beginn der Minimierung oftmals gute Fortschritte macht und vergleichsweise günstig ist, schlagen wir folgendes Vorgehen bei der Approximation der Optimallösung von gegebenen Programmen (P) und (D) (welche zumindest Voraussetzung 1 erfüllen) vor:

1. Minimiere $\tilde{\phi}$ mit L-BFGS um sich der Optimallösung ausreichend anzunähern.
2. Minimiere danach $\psi = \tilde{\phi} + \tilde{f}$ mit dem verallgemeinerten Newton-Verfahren um die Optimallösung zu bestimmen.

Bei der Implementierung kann ähnlich wie in Abschnitt 2.6 beschrieben von Anfang an die Funktion $\psi_\alpha := \tilde{\phi} + \alpha \tilde{f}$ mit einem $\alpha \geq 0$ minimiert werden. Dieser Parameter kann anhand gewisser Kriterien erhöht bzw gesenkt werden. Wenn die Konvergenz des zu Beginn verwendeten L-BFGS-Verfahrens zu langsam wird, erfolgt ein Wechsel zum verallgemeinerten Newton-Verfahren.

Die gemachten Tests zeigten, dass sich die APD-Methode unter Verwendung des verallgemeinerten Newton-Verfahrens zur Bestimmung der Suchrichtung (wobei

das Newton-System mit dem cg-Steihaug-Algorithmus bis zu einer festgesetzten Genauigkeit gelöst wurde) global sehr ähnlich zur L-BFGS-Variante verhält. Da die Berechnung der Newton-Suchrichtung jedoch viel teurer als ein L-BFGS-Schritt ist, sollte die L-BFGS-Variante möglichst lange verwendet werden. Das verallgemeinerte Newton-Verfahren hat zwar einen größeren Konvergenzbereich als L-BFGS, aber wenn man zu weit von der Optimallösung entfernt startet, dann kann es passieren, dass die Minimierung mittels L-BFGS sehr langsam wird, bevor die Verwendung der Newton-Variante sinnvoll ist.

Im nächsten Abschnitt wird ein anderer Zugang beschrieben, welcher mit dem APD-Verfahren kombiniert werden kann.

3.2 AHO-QMR

3.2.1 Motivation

Wir werden uns in diesem Abschnitt teilweise von dem bisherigen Vorgehen lösen. Wenn es um hohe Genauigkeit geht, dann ist die Minimierung der Funktionen $\tilde{\phi}$ (und $\tilde{f}$) insofern problematisch, da die mit dem Operator $\mathcal{A}$ (siehe Abschnitt 2.1) zusammenhängenden Rechenfehler bei den Projektionen auf $\mathbf{L}$ und $\mathbf{A}$ quadriert werden können. Bei einer gegeben Rechengenauigkeit kann es dadurch schwierig werden, den Approximationsfehler bei der Bestimmung der Optimallösung von gegebenen Programmen (P) und (D) unter eine gewisse Schranke zu drücken. Wir werden nun ein Verfahren vorstellen, in dem dieses Problem vermieden wird. Dazu betrachten wir zunächst ein System, welches aus den Innere-Punkte-Verfahren bekannt ist. Für lineare Programme von der Form (P) und (D) hat das System beispielsweise die folgende Form:

$$\begin{aligned} Ax &= \bar{b}, \\ A^T y + s &= c, \\ x \circ s &= 0, \\ x \geq 0, \quad s &\geq 0, \end{aligned} \tag{3.2}$$

wobei für zwei Elemente $V, W \in I\!R^{m \times n}$

$$V \circ W := (V_{ij} W_{ij})_{\substack{i=1,\ldots,m, \\ j=1,\ldots,n}}$$

gilt. Ist (x, y, s) eine Lösung des Systems (3.2), dann ist das Tripel nach Korollar 1 und einer zu (2.1) äquivalenten Formulierung für lineare Proggramme eine Optimallösung für (P) und (D). Als „zentraler Pfad" werden alle Punkte $(x(\mu), y(\mu), s(\mu))$ bezeichnet, die das folgende System für ein $\mu > 0$ lösen:

$$\begin{aligned} Ax &= \bar{b}, \\ A^T y + s &= c, \\ x \circ s &= \mu e, \\ x \geq 0, \quad s &\geq 0, \end{aligned} \tag{3.3}$$

wobei e ein Vektor entsprechender Dimension ist, dessen Komponenten alle 1 sind.
Wir fassen kurz die Funktionsweise der Innere-Punkte-Methoden zur Lösung von (P) und (D) zusammen ohne auf die benötigten Voraussetzungen einzugehen, wobei wir anmerken, dass Voraussetzung 2 hinreichend für die Existenz des „zentralen Pfades" ist.

1. Für ein $\mu_k > 0$ sei ein Punkt (x_k, y_k, s_k) mit $x_k > 0$ und $s_k > 0$ gegeben, welcher in einer Umgebung von $(x(\mu_k), y(\mu_k), s(\mu_k))$ liegt. Durch einen Newton-Schritt zur Lösung des Systems (3.3) ausgehend von der Stelle (x_k, y_k, s_k) erhält man die Suchrichtung $(\Delta x_k, \Delta y_k, \Delta s_k)$. Man wählt $\alpha_k^x, \alpha_k^s \in (0, 1]$ möglichst groß, so dass jedoch $x_{k+1} := x_k + \alpha_k^x \Delta x_k > 0$ und $s_{k+1} := s_k + \alpha_k^s \Delta s_k > 0$ gilt uns setzt $y_{k+1} := y_k + \alpha_s \Delta y_k$.
2. Man setzt $\mu_{k+1} := \beta \mu_k$ für ein $\beta \in (0, 1)$ und geht wieder zu Schritt 1.

Auf diese Weise versucht man sich schrittweise der Lösung des Systems (3.2) zu nähern. Die Voraussetzungen unter denen das Verfahren konvergiert und weitere interessante Eigenschaften von Innere-Punkte-Methoden können z.B. in [10] nachgelesen werden.

Für semidefinite Programme lassen sich analog verschiedene ähnliche Formulierungen zu (3.2) und (3.3) herleiten, mit deren Hilfe man die Optimallösungen der Probleme (P) und (D) bestimmen kann. Im Gegensatz zu linearen Programmen kann man insbesondere die dritte Gleichung beider Systeme auf viele Arten ausdrücken, die zueinander äquivalent sind und die Verwendung eines Innere-Punkte-Verfahrens, welches ebenfalls das Newton-Verfahren in Verbindung mit der Verkleinerung des Parameters μ nutzt, erlauben. Eine Übersicht dieser Verfahren findet sich z.B. in [10] und [19]. Wir werden uns aber mit einer speziellen Variante dieser Verfahrensklasse beschäftigen und darauf eingehen, wie diese Variante mit der in Abschnitt 1.3 eingeführten APD-Methode verknüpft werden kann um Optimallösungen hoher Genauigkeit für Probleme von hoher Dimension zu bestimmen.

3.2.2 Das AHO System

Das Bestimmen der Optimallösung von gegebenen primal-dualen Programmen (P) und (D) ist äquivalent zur Lösung des Systems

$$\begin{aligned} \mathcal{A}(X) &= \bar{b}, \\ \mathcal{A}^*(y) + S &= C, \\ XS + SX &= 0, \\ X \succeq 0, \quad S &\succeq 0. \end{aligned} \tag{3.4}$$

Die letzte Gleichung folgt dabei aus (2.1), da für $X \succeq 0$, $S \succeq 0$ die folgende Äquivalenzkette gilt: $X \bullet Y \Leftrightarrow XS = 0 \Leftrightarrow SX = 0 \Leftrightarrow XS + SX = 0$. Dieses

System heißt AHO-Symmetrisierung (Alizadeh, Haeberly, and Overton [1]) und gehört zu einem speziellen Innere-Punkte-Verfahren. Um eine Lösung von (3.4) zu approximieren, bestimmt man ausgehend von einem gegebenen Punkt (X, y, S) einen Suchschritt $(\Delta X, \Delta y, \Delta S)$, welcher das folgende lineare Gleichungssystem löst:

$$\begin{array}{rrrcl} \mathcal{A}(\Delta X) & & & = & \bar{b} - \mathcal{A}(X), \\ & \mathcal{A}^*(\Delta y) \quad +\Delta S & & = & C - \mathcal{A}^*(y) - S, \\ S\Delta X + \Delta X S & & +X\Delta S + \Delta S X & = & -XS - SX. \end{array} \tag{3.5}$$

Unter gewissen Voraussetzungen (siehe z.B. [10]) ist Existenz und Eindeutigkeit der Lösung von (3.5) gegeben, so dass diese einem Newton-Schritt zur Bestimmung der Lösung von (3.4) entspricht. Die Vorgehensweise ist gleich, wenn die rechte Seite der dritten Gleichung von (3.4) um den Summanden μI für ein $\mu > 0$ ergänzt wird um somit eine Annäherung an den „zentralen Pfad" zu ermitteln. Hierbei gilt typischerweise $X \succ 0$, $S \succ 0$. Durch eine Schrittweitenanpassung $\alpha_X, \alpha_S \in (0, 1]$ wird zusätzlich noch $X + \alpha_X \Delta X \succ 0$ und $S + \alpha_S \Delta S \succ 0$ gewährleistet.

Voraussetzung 4. *Der zum Programm (P) bzw. $(\bar{P})$ gehörende Operator $\mathcal{A}$ besitzt vollen Rang, d.h. die Matrizen $A^{(1)}, \ldots, A^{(m)}$ sind linear unabhängig.*

Die AHO-Suchrichtung ist von besonderem Interesse. Für eine Reihe von Innere-Punkte-Verfahren, darunter die AHO-Variante, wurde unter annehmbaren Voraussetzungen superlineare Konvergenz nachgewiesen. Sind Voraussetzungen 3 und 4 erfüllt, dann konvergiert das Newton-Verfahren zur Lösung des Systems (3.4) in einer Umgebung der Optimallösung $(X^{opt}, y^{opt}, S^{opt})$ sogar quadratisch, siehe [1]. Diese Aussage wollen wir in dem Verfahren, welches weiter unten beschrieben wird, nutzen.

Sind n und m sehr groß, dann ist eine direkte Lösung des Systems (3.5) nicht möglich. Für diesen Fall stellen wir im Folgenden eine Reformulierung der AHO-Linearisierung (3.5) vor, die mit iterativen Methoden gelöst werden kann.

3.2.3 Startpunkt

Wir gehen im Folgenden davon aus, dass zunächst das APD-Verfahren genutzt wird um sich einer Optimallösung (X^{opt}, S^{opt}) ausreichend zu nähern. Dies kann z.B. mittels des in Abschnitt 2.5 vorgestellten L-BFGS-Verfahrens geschehen. Wir nehmen weiterhin an, dass Voraussetzungen 3 und 4 erfüllt sind.

Sei $k \in \{1, \ldots, n-1\}$ die Anzahl der positiven Eigenwerte von X^{opt} und m die Anzahl der linearen Nebenbedingungen (in $\mathcal{A}$). Die Eindeutigkeit von (X^{opt}, S^{opt})

impliziert[1] $m \in \{\frac{k(k+1)}{2}, \ldots, \frac{k(k+1)}{2} + k(n-k)\}$. Ist andererseits m gegeben, dann folgt

$$k \in \{\lceil \frac{1}{2}(2n+1-\sqrt{(2n+1)^2-8m})\rceil, \ldots, \lfloor \frac{1}{2}(\sqrt{1+8m}-1)\rfloor\}. \tag{3.6}$$

Wir nehmen für die folgende Überlegung an, dass eine durch das APD-Verfahren berechnete Approximation (X^{APD}, S^{APD}) von (X^{opt}, S^{opt}) gegeben ist.

Zuerst wird ein korrigiertes Paar $(X^C, S^C) \in \mathcal{S}^n_+ \times \mathcal{S}^n_+$ bestimmt, welches eine gemeinsame Eigenbasis besitzt. Dabei wollen wir die Tatsache ausnutzen, dass Eigenräume zu verschiedenen Eigenwerten lokal Lipschitz-stetig sind. Falls also X^{APD} und S^{APD} fast positiv semidefinit und strikt komplementär sind, dann erlaubt uns diese Eigenschaft die Eigenvektoren von X^{APD} und von S^{APD} zu trennen, indem wir die Differenz $Z := X^{APD} - S^{APD}$ bilden. Die Bestimmung des Startpunktes wird folgendermaßen ausgeführt:

1. Berechne $(X_P, S_P) := (\Pi_{\mathcal{S}^n_+}(X^{APD}), \Pi_{\mathcal{S}^n_+}(S^{APD}))$

2. Setze $Z := \frac{X_P}{\|X_P\|_F} - \frac{S_P}{\|S_P\|_F}$ und bestimme die Eigenwertzerlegung $Z = UDU^T$.

3. Setze $X^e := U^T X_P U$, $S^e := U^T S_P U$
   und $\Lambda := \mathrm{Diag}(\mathrm{diag}(X^e))$, $\Sigma := \mathrm{Diag}(\mathrm{diag}(S^e))$.

4. Definiere $\delta := (\|X^e - \Lambda\|_F^2 + \|S^e - \Sigma\|_F^2)^{1/2}$.

   Erhöhe, falls nötig, die Diagonaleinträge von Λ und Σ, so dass für jeden Index i entweder $(\Lambda)_{i,i} \geq \delta$ oder $(\Sigma)_{i,i} \geq \delta$ erfüllt ist, und so dass die Anzahl der Diagonaleinträge in Λ (und in Σ), die einen Wert von mindestens δ haben, mit Bedingung (3.6) vereinbar ist.

5. Setze $X^C := U\Lambda U^T$ und $S^C := U\Sigma U^T$.

Bemerkung: Da die Optimallösung (X^{opt}, S^{opt}) ein Fixpunkt des kontrahierenden Operators $\Pi_{\mathcal{S}^n_+ \times \mathcal{S}^n_+}$ ist, ist der Punkt $(X_P, S_P) = (\Pi_{\mathcal{S}^n_+}(X^{APD}), \Pi_{\mathcal{S}^n_+}(S^{APD}))$ näher an (X^{opt}, S^{opt}) als (X^{APD}, S^{APD}). Das Maß δ ist eine obere Schranke für die Distanz von (X_P, S_P) zu einem Paar in $\mathcal{S}^n_+ \times \mathcal{S}^n_+$, welches eine gemeinsame Eigenbasis besitzt. Durch die Modifikation der Diagonaleinträge von Λ und Σ in Schritt 4 kann die Distanz von (X^C, S^C) zu (X_P, S_P) aber größer als δ sein.

[1] Besitzt X^{opt} insgesamt k positive Eigenwerte, dann bleibt der zugehörige $k \times k$-Block von X^{opt} durch belieibige kleine Störungen positiv definit und die veränderte Matrix strikt komplementär zu S^{opt}. Die Eindeutigkeit des $k \times k$-Blocks impliziert $\frac{k(k+1)}{2} \leq m$. Mit der gleichen Argumentation für S^{opt} erhält man $m \leq \frac{k(k+1)}{2} + k(n-k)$.

Wenn eine gemeinsame Eigenbasis gegeben ist, dann ist eine „ausreichende" Anzahl an Eigenwerten von X^C and S^C, die einen Wert von mindestens δ aufweisen, und die Bedingung $X^C + S^C \succeq \delta I$ ausschlaggebend dafür, dass die Umformulierungen im nächsten Abschnitt nicht von Inversen abhängen, die schlecht konditioniert sind (also zu großen Rundungsfehlern bei der Auswertung führen können).

Falls (X^{APD}, S^{APD}) nah genug an einer strikt komplementären Optimallösung (X^{opt}, S^{opt}) liegt, dann liegt auch der korrigierte Punkt (X^C, S^C) in der Nähe von (X^{opt}, S^{opt}). Dies wurde in den Tests beobachtet, deren Resultate in Abschnitt 3.2.6 präsentiert werden.

3.2.4 Symmetrisierung des AHO Systems

Als Nächstes werden wir eine Umformulierung der AHO-Linearisierung (3.5) beschreiben. Wir nehmen an, dass ein gegebenes Paar $(X, S) = (X^C, S^C)$ die Ausgabe der in Abschnitt 3.2.3 veranschaulichten Prozedur ist.

Sei eine entsprechende Matrix U gegeben, die $X = U\Lambda U^T$ und $S = U\Sigma U^T$ mit Diagonalmatrizen $\Lambda,\ \Sigma$ erfüllt. Der Operator $\mathcal{U} : \mathcal{S}^n \to \mathcal{S}^n$, definiert durch $\mathcal{U}(Y) := UYU^T$, ist eine bijektive lineare Transformation. Sei $\Delta\hat{X} := U^T\Delta X U$, $\Delta\hat{S} := U^T\Delta S U$, $\hat{C} := U^T C U$. Sei der lineare Operator $\hat{\mathcal{A}}$ definiert durch

$$\hat{\mathcal{A}}(\hat{X}) := \begin{pmatrix} \hat{A}^{(1)} \bullet \hat{X} \\ \vdots \\ \hat{A}^{(m)} \bullet \hat{X} \end{pmatrix} := \begin{pmatrix} U^T A^{(1)} U \bullet \hat{X} \\ \vdots \\ U^T A^{(m)} U \bullet \hat{X} \end{pmatrix}.$$

Durch Verwendung der Transformation $\mathcal{U}$ erhält man die folgende Umformulierung des Systems (3.5):

$$(T_1) \qquad \begin{array}{rrrcl} \hat{\mathcal{A}}(\Delta\hat{X}) & & & = & \bar{b} - \mathcal{A}(X), \\ & \hat{\mathcal{A}}^*(\Delta y) \quad +\Delta\hat{S} & & = & \hat{C} - \hat{\mathcal{A}}^*(y) - \Sigma, \\ \Sigma\Delta\hat{X} + \Delta\hat{X}\Sigma & & +\Lambda\Delta\hat{S} + \Delta\hat{S}\Lambda & = & -2\Lambda\Sigma. \end{array}$$

Nach Definition gilt $\mathcal{A}(\Delta X) = \hat{\mathcal{A}}(\Delta\hat{X})$.
Seien die Einträge der symmetrischen $n \times n$-Matrizen $\tilde{\Lambda}$, $\tilde{\Sigma}$ für $1 \le i, j \le n$ durch $\tilde{\Lambda}_{i,j} := \Lambda_i + \Lambda_j$ und $\tilde{\Sigma}_{i,j} := \Sigma_i + \Sigma_j$ gegeben. Dann kann die letzte Gleichung von (T_1) zu

$$\tilde{\Sigma} \circ \Delta\hat{X} + \tilde{\Lambda} \circ \Delta\hat{S} = -2\Lambda\Sigma$$

umformuliert werden.

In obenstehender Formulierung kommen alle Gleichungen, die sich nicht auf die Diagonalelemente beziehen, doppelt vor. Diese Redundanz wird in der folgenden äquivalenten Formulierung (T_2) eliminiert. Zur einfacheren Darstellung

wird sowohl der in Abschnitt 2.5 definierte Operator svec als auch die folgende Abbildung verwendet:

$$\overline{\mathsf{vec}} : \mathcal{S}^n \to I\!R^{\frac{n(n+1)}{2}},\ \overline{\mathsf{vec}}(X) = (X_{ij})_{\substack{i=1\ldots n,\\ j=i\ldots n}}.$$

Setze $\hat{\Lambda} := \overline{\mathsf{vec}}(\tilde{\Lambda})$, $\hat{\Sigma} := \overline{\mathsf{vec}}(\tilde{\Sigma})$, $\Delta\hat{x} := \mathsf{svec}(\Delta\hat{X})$, $\Delta\hat{s} := \mathsf{svec}(\Delta\hat{S})$ und

$$\hat{\Lambda}^D := \mathsf{Diag}(\hat{\Lambda}), \quad \hat{\Sigma}^D := \mathsf{Diag}(\hat{\Sigma}), \quad \hat{A} := \mathsf{svec}(\hat{\mathcal{A}}) := \begin{bmatrix} \mathsf{svec}(\hat{A}^{(1)})^T \\ \vdots \\ \mathsf{svec}(\hat{A}^{(m)})^T \end{bmatrix},$$

so dass $\hat{\mathcal{A}}(\Delta\hat{X}) = \hat{A}\Delta\hat{x}$ erfüllt ist.

System (T_1) kann dann folgendermaßen umformuliert werden:

$$(T_2) \qquad \begin{bmatrix} & \hat{A} & \\ \hat{A}^T & & I \\ & \hat{\Sigma}^D & \hat{\Lambda}^D \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\hat{x} \\ \Delta\hat{s} \end{bmatrix} = \begin{bmatrix} \bar{b} - \mathcal{A}(X) \\ \mathsf{svec}(\hat{C} - \hat{\mathcal{A}}^*(y) - \Sigma) \\ -2\mathsf{svec}(\Lambda\Sigma) \end{bmatrix} =: \begin{bmatrix} \mathrm{rhs}^1 \\ \mathrm{rhs}^2 \\ \mathrm{rhs}^3 \end{bmatrix}.$$

Als Nächstes eliminieren wir einen Teil der Variable $\Delta\hat{s}$. Dazu wählen wir zunächst die m größten Werte von $\{\frac{\hat{\Lambda}_d}{\hat{\Sigma}_d} \mid 1 \le d \le \frac{n(n+1)}{2}\}$, wobei wir $\frac{\hat{\Lambda}_d}{0} := \infty$ setzen, und fassen die zu diesen Werten gehördenden Indizes d als Menge $\mathtt{I}$ zusammen. Es kann keinen Index d geben, so dass $\frac{\hat{\Lambda}_d}{\hat{\Sigma}_d} = \frac{0}{0}$ gilt. Sonst würde ein Paar (i,j) existieren, so dass $\tilde{\Lambda}_{ij} = \Lambda_{ii} + \Lambda_{jj} = 0$, $\tilde{\Sigma}_{ij} = \Sigma_{ii} + \Sigma_{jj} = 0$ gilt. Dies widerspricht allerdings der Voraussetzung $X + S \succ 0 \Leftrightarrow \Lambda + \Sigma \succ 0$.
Entsprechend Schritt 4 in Abschnitt 3.2.3 existiert ein

$$\tilde{k} \in \{\lceil \frac{1}{2}(2n+1-\sqrt{(2n+1)^2-8m})\rceil, \ldots, \lfloor \frac{1}{2}(\sqrt{1+8m}-1)\rfloor\}$$

und $q_1, \ldots, q_n \in \{1, \ldots, n\}$, $q_\mu \neq q_\nu$ für $\mu \neq \nu$, so dass $\Lambda_{q_1q_1}, \ldots, \Lambda_{q_{\tilde{k}}q_{\tilde{k}}} \ge \delta$ und $\Sigma_{q_{\tilde{k}+1}q_{\tilde{k}+1}}, \ldots, \Sigma_{q_nq_n} \ge \delta$ gilt. Somit existieren mindestens $\frac{\tilde{k}(\tilde{k}+1)}{2} + \tilde{k}(n-\tilde{k}) \ge m$ Elemente $d \in \{1, \ldots, \frac{n(n+1)}{2}\}$, welche $\hat{\Lambda}_d > 0$ erfüllen. Damit gilt $\hat{\Lambda}_d > 0$ für alle $d \in \mathtt{I}$. Weiterhin existieren mindestens $\frac{n(n+1)}{2} - m$ Elemente $d \in \{1, \ldots, \frac{n(n+1)}{2}\}$, welche $\hat{\Sigma}_d > 0$ erfüllen. Nach Definition der Menge $\mathtt{I}$ gehört damit jeder Index d mit $\hat{\Sigma}_d = 0$ zur Menge $\mathtt{I}$. Es folgt somit, dass die Festsetzung

$$\Delta\hat{s}_d = \frac{\mathrm{rhs}^3_d - \hat{\Sigma}_d\Delta\hat{x}_d}{\hat{\Lambda}_d} \tag{3.7}$$

für alle $d \in \mathtt{I}$ wohldefiniert ist. Mit der Definition $\mathtt{II} := \{1, \ldots, \frac{n(n+1)}{2}\} \setminus \mathtt{I}$ folgt weiterhin

$$\hat{\Sigma}_d > 0 \ \forall\ d \in \mathtt{II}. \tag{3.8}$$

Wir „zerlegen" (T_2) in das System

$$(T_2') \qquad \begin{bmatrix} & \hat{A}_{\mathtt{I}} & \hat{A}_{\mathtt{II}} & & \\ \hat{A}_{\mathtt{I}}^T & & & I & \\ \hat{A}_{\mathtt{II}}^T & & & & I \\ & \hat{\Sigma}_{\mathtt{I}}^D & & \hat{\Lambda}_{\mathtt{I}}^D & \\ & & \hat{\Sigma}_{\mathtt{II}}^D & & \hat{\Lambda}_{\mathtt{II}}^D \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\hat{x}_{\mathtt{I}} \\ \Delta\hat{x}_{\mathtt{II}} \\ \Delta\hat{s}_{\mathtt{I}} \\ \Delta\hat{s}_{\mathtt{II}} \end{bmatrix} = \begin{bmatrix} \mathrm{rhs}^1 \\ \mathrm{rhs}_{\mathtt{I}}^2 \\ \mathrm{rhs}_{\mathtt{II}}^2 \\ \mathrm{rhs}_{\mathtt{I}}^3 \\ \mathrm{rhs}_{\mathtt{II}}^3 \end{bmatrix}.$$

Unter Verwendung von (3.7) eliminieren wir $\Delta\hat{s}_{\mathtt{I}}$ aus (T_2') und erhalten

$$(T_3) \qquad \begin{bmatrix} & \hat{A}_{\mathtt{I}} & \hat{A}_{\mathtt{II}} & \\ \hat{A}_{\mathtt{I}}^T & Q_{\mathtt{I}} & & \\ \hat{A}_{\mathtt{II}}^T & & & I \\ & & \hat{\Sigma}_{\mathtt{II}}^D & \hat{\Lambda}_{\mathtt{II}}^D \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\hat{x}_{\mathtt{I}} \\ \Delta\hat{x}_{\mathtt{II}} \\ \Delta\hat{s}_{\mathtt{II}} \end{bmatrix} = \begin{bmatrix} \mathrm{rhs}^1 \\ \mathrm{rhs}_{\mathtt{I}}^2 - \widetilde{\mathrm{rhs}}_{\mathtt{I}}^3 \\ \mathrm{rhs}_{\mathtt{II}}^2 \\ \mathrm{rhs}_{\mathtt{II}}^3 \end{bmatrix}.$$

Hierbei gilt $Q_{\mathtt{I}} := -(\hat{\Lambda}_{\mathtt{I}}^D)^{-1}\hat{\Sigma}_{\mathtt{I}}^D$ und $\widetilde{\mathrm{rhs}}_{\mathtt{I}}^3 := (\frac{\mathrm{rhs}_d^3}{\hat{\Lambda}_d})_{d\in\mathtt{I}}$.

Der letzte Schritt ist die Symmetrisierung des Systems (T_3):
Aus (3.8) folgt zunächst, dass die Elemente $Q_{\mathtt{II}} := (\hat{\Sigma}_{\mathtt{II}}^D)^{-1}\hat{\Lambda}_{\mathtt{II}}^D$ und $\widetilde{\mathrm{rhs}}_{\mathtt{II}}^3 := (\frac{\mathrm{rhs}_d^3}{\hat{\Sigma}_d})_{d\in\mathtt{II}}$ wohldefiniert sind. Somit ist die Formulierung

$$(T_4) \qquad \begin{bmatrix} & \hat{A}_{\mathtt{I}} & \hat{A}_{\mathtt{II}} & \\ \hat{A}_{\mathtt{I}}^T & Q_{\mathtt{I}} & & \\ \hat{A}_{\mathtt{II}}^T & & & I \\ & & I & Q_{\mathtt{II}} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\hat{x}_{\mathtt{I}} \\ \Delta\hat{x}_{\mathtt{II}} \\ \Delta\hat{s}_{\mathtt{II}} \end{bmatrix} = \begin{bmatrix} \mathrm{rhs}^1 \\ \mathrm{rhs}_{\mathtt{I}}^2 - \widetilde{\mathrm{rhs}}_{\mathtt{I}}^3 \\ \mathrm{rhs}_{\mathtt{II}}^2 \\ \widetilde{\mathrm{rhs}}_{\mathtt{II}}^3 \end{bmatrix}$$

äquivalent zu (T_3). Das erhaltene System (T_4) ist symmetrisch und von der Dimension $n(n+1)$.

Die Diagonalmatrizen $Q_{\mathtt{I}}$ und $Q_{\mathtt{II}}$ enthalten jeweils nur die Inversen der „großen" Komponenten von $\tilde{\Lambda}$ und $\tilde{\Sigma}$. Es ist ohne Weiteres möglich, zusätzlich die Komponenten $\Delta\hat{x}_{\mathtt{II}}$ oder $\Delta\hat{s}_{\mathtt{II}}$ durch Verwendung des dritten oder vierten Zeilenblocks zu eliminieren. Eine stabile Symmetrisierung des resultierenden Systems ist aber schwer zu erstellen.

Die Matrix $\hat{A}$ im System (T_4) ist normalerweise voll besetzt und wird daher nicht explizit berechnet. Es werden nur dem System (T_4) „entsprechende" Multiplikationen ausgeführt. Eine Multiplikation der Form $\hat{A}\Delta\hat{x}$ wird beispielsweise zunächst durch Transformation der Matrixvariable $\Delta\hat{X}$ mit U und U^T und anschließend durch Verwendung der Matrix

$$A := \mathsf{svec}(\mathcal{A}) := \begin{bmatrix} \mathsf{svec}(A^{(1)})^T \\ \vdots \\ \mathsf{svec}(A^{(m)})^T \end{bmatrix}$$

berechnet.

3.2.5 Präkonditionierung des symmetrischen Systems

Einer der wesentlichen Bestandteile der APD-Methode ist die Matrix AA^T und ihre Verwendung in der Projektion auf den Raum $\mathbf{A}$. Ein Cholesky Faktor für diese Matrix muss für das gesamte Verfahren nur ein einziges mal berechnet werden. Wenn A dünn besetzt ist, dann ist die Benutzung des Cholesky Faktors von AA^T i.Allg. sehr viel günstiger als die Faktorisierungen, die bei Innere-Punkte-Methoden in jeder Iteration ausgeführt werden um einen neuen Suchschritt zu bestimmen. In den numerischen Beispielen, die im Folgenden präsentiert werden, wird der für die APD-Methode berechnete Cholesky Faktor erneut verwendet um den ersten Zeilen- und den ersten Spaltenblock von System (T_4) umzuformulieren (Präkonditionierung). Damit erhält man einen orthogonalen Zeilen- und Spaltenblock. Das entstehende System wird – wie bereits erwähnt – nicht explizit berechnet und gespeichert.

Das resultierende präkonditionierte System wurde mit einer symmetrischen Version des QMR-Verfahrens gelöst. Das ursprüngliche QMR-Verfahren wird in [7, 17] genau erläutert. Wir geben im Folgenden die symmetrische Version an:

Algorithmus 5 (Symmetrisches QMR-Verfahren)**.** *Gesucht ist die Lösung x der Gleichung $Mx = w$ mit $M = M^T \in \mathbb{R}^{n\times n}$, $x, w \in \mathbb{R}^n$.*

Gegeben sei ein Startpunkt x^0 und eine Fehlertoleranz $\varepsilon > 0$.
Berechne $r_0 := w - Mx^0$, $\xi := \xi_0 := \|r_0\|_2$. Setze $v := 0 \in \mathbb{R}$, $v_{neu} := r_0/\xi_0$, $\delta_{neu} := 0 \in \mathbb{R}$ und weiterhin $\mu := \mu_{neu} := 1$, $\nu := \nu_{neu} := 0 \in \mathbb{R}$, $p_{neu} := p := 0 \in \mathbb{R}^n$.

Solange $|\xi|/\xi_0 > \varepsilon$

1. *Setze $\delta := \delta_{neu}$, $v_{alt} := v$ und danach $v := v_{neu}$.*
2. *Berechne $\alpha := v^T Mv$ und weiterhin $\tilde{v}_{neu} := Mv - \alpha v - \delta v_{alt}$.*
3. *Berechne $\delta_{neu} := \|\tilde{v}_{neu}\|_2$ und anschließend $v_{neu} = \tilde{v}_{neu}/\delta_{neu}$.*
4. *Setze $\mu_{alt} := \mu$, danach $\mu := \mu_{neu}$, $\nu_{alt} := \nu$ und schließlich $\nu := \nu_{neu}$.*
5. *Berechne $p_1 := \nu_{alt}\delta$, $p_2 := \mu\mu_{alt}\delta + \nu\alpha$. $\tilde{p}_3 := \mu\alpha - \nu\mu_{alt}\delta$.*
6. *Berechne $p_3 := \sqrt{\tilde{p}_3^2 + {\delta_{neu}}^2}$.*
7. *Setze $\mu_{neu} := \tilde{p}_3/p_3$, $\nu_{neu} := \delta_{neu}/p_3$.*
8. *Setze $p_{alt} := p$, dann $p := p_{neu}$ und anschließend $p_{neu} := (v - p_1 p_{alt} - p_2 p)/p_3$.*
9. *Berechne $x_{neu} := x + \mu_{neu}\xi p_{neu}$ und $\xi_{neu} := -\nu_{neu}\xi$.*
10. *Setze $x := x_{neu}$ und $\xi := \xi_{neu}$.*

Die Berechnung eines AHO-Suchschrittes ist äquivalent zur Lösung der Präkonditionierung von System (T_4). Mit dem oben vorgestellten Algorithmus 5 kann die Lösung bis zu einer gewünschten Genauigkeit approximiert werden. Für sehr hohe Dimensionen n bietet dieser Umweg eine Möglichkeit den Suchschritt in annehmbarer Zeit zu berechnen. Insbesondere ist der Speicherbedarf nur unwesentlich höher als der des APD-Verfahrens. Für semidefinite Programme (P) und (D) der Dimension n (d.h. $X, S \in \mathcal{S}^n$) und das dazugehörige AHO-System liegt der Rechenaufwand pro QMR-Iteration in $\mathcal{O}(mn^2)$. Ist der Operator $\mathcal{A}$ dünn besetzt, dann liegt der Aufwand pro Iteration typischerweise in $\mathcal{O}(n^3)$.

In der Implementierung, deren Ergebnisse weiter unten präsentiert werden, wurde die Genauigkeit ε in Abhängigkeit vom Anfangsfehler r_0 gewählt. Je kleiner r_0 war, desto höher war die bei der Berechnung des AHO-Schrittes geforderte Genauigkeit.

Wir haben in diesem Abschnitt eine Berechnung des AHO-Suchschrittes mit Hilfe des symmetrischen QMR-Verfahrens vorgestellt. Ausgehend von einem Paar $(X^{APD}, S^{APD}) \in \mathbf{A}$ wird ein neues Paar (X^C, S^C) (Abschnitt 3.2.3) und eine Suchrichtung $(\Delta X^C, \Delta S^C) \in \mathcal{S}^n \times \mathcal{S}^n$ (die AHO-QMR-Suchrichtung) bestimmt. Eine anschließende Projektion der neuen Iterierten auf den Raum $\mathbf{A}$ erzeugt eine neue Approximation an die Optimallösung:

$$(X_+^{APD}, S_+^{APD}) := \Pi_{\mathbf{A}}((X^C, S^C) + (\Delta X^C, \Delta S^C)) \in \mathbf{A}.$$

Diese Prozedur wird im Folgenden als AHO-QMR-Schritt bezeichnet.

In den meisten Tests konvergierte die AHO-QMR-Methode sehr schnell, falls das Ausgangspaar (X, S) bereits in der Nähe von (X^{opt}, S^{opt}) lag. Ist man von der Optimallösung noch weit entfernt, dann konvergiert das AHO-QMR-Verfahren möglicherweise gar nicht oder gegen ein Paar, welches nicht in $\mathcal{S}_+^n \times \mathcal{S}_+^n$ liegt. Daher ist es sinnvoll, die AHO-QMR-Methode mit dem APD-Verfahren zu kombinieren:

1. Verwende das APD-Verfahren zur Minimierung der Funktion $\tilde{\phi} + \alpha \tilde{f}$, siehe dazu 2.6.

2. Wechsle zu AHO-QMR, d.h. löse das präkonditionierte System (T_4) mit dem symmetrischen QMR-Verfahren. Wiederhole AHO-QMR solange es effektiver als APD ist. Gehe zurück zur APD-Methode, falls nötig, d.h. gehe zu Schritt 1, falls der AHO-QMR-Schritt gewisse Konvergenzbedingungen nicht erfüllt.

Im Folgenden werden wir diese Kombination als HYBRID-Verfahren bezeichnen.

Kondition

Die Normalengleichungen, die in Innere-Punkte-Methoden zur Bestimmung der Suchrichtung gelöst werden müssen, sind i.Allg. schlecht konditioniert. Dennoch

führt die direkte Faktorisierung dieser Normalengleichungen oft zu guten numerischen Ergebnissen. Die Effektivität von iterativen Methoden zur Lösung dieser Gleichungen ist allerdings stark von der Kondition abhängig. Die Definition von (X^C, S^C) in Abschnitt 3.2.3 impliziert jedoch, dass die Umformulierungen zum System (T_4) die Kondition des Ausgangssystems (3.5) nicht systematisch verschlechtern. Durch die Erstellung eines symmetrischen Systems kann die Anzahl der zu einer ausreichenden Approximation des AHO-Schrittes führenden QMR-Schritte gesenkt werden. Durch die Ausnutzung der Symmetrie kann der Rechenaufwand im Vergleich zum normalen QMR-Verfahren mindestens halbiert werden.

Bevor die APD- bzw. die HYBRID-Methode zur Lösung von semidefiniten Programmen gestartet wurde, sind die Probleme zunächst äquivalent umskaliert worden – siehe Abschnitt 2.6.

Diese Umskalierung impliziert $\|X\|_F \geq 1$ und $\|S\|_F \geq 1$ für jedes Paar $(X, S) \in \mathbf{A}$. Wegen dieser Eigenschaft und der Tatsache, dass die Iterierten (X, S) nur die Kegelbedingung verletzen, kann der Fehler der APD- und der HYBRID-Methode durch das Maß $|\lambda_{min}(X)| + |\lambda_{min}(S)|$ gemessen werden. Dabei sind (X, S) die Iterierten bzgl. der zu (P) und (D) gehörenden umskalierten Daten. $\lambda_{min}(X)$ bezeichnet den kleinsten Eigenwert von X und analog für S. Dieses Fehlermaß wird im zweiten und den folgenden Abschnitten der nächsten Sektion verwendet, da dort nur die APD- und HYBRID-Methode betrachtet werden.

3.2.6 Numerische Ergebnisse

Um das Potential der HYBRID-Methode zu illustrieren präsentieren wir im Folgenden einige numerische Ergebnisse. Wie in Abschnitt 2.6 wurde auch hier wieder eine MATLAB® Implementierung verwendet.

Vergleich mit SeDuMi

Zunächst wurde HYBRID mit dem Software-Paket SeDuMi verglichen. Eine Beschreibung dieses Pakets findet sich z.B. in [24, 18]. Es wurden einige dünnbesetzte semidefinite Programme mit einer Dimension bis zu `n=300` getestet. Semidefinite Programme mit einer Dimension von 400 oder höher konnten auf dem TEST-PC mit SeDuMi nicht berechnet werden. Die Beispiele wurden mit einem Generator für zufällige dünnbesetzte semidefinite Programme der Forschungsgruppe von Franz Rendl erzeugt, siehe [22]. Die Eingabe für diesen Generator besteht aus der Dimension der Matrixvariable `n`, der Anzahl der linearen Restriktionen `m`, einem „Dünnbesetztheitsparameter" `p`, der besagt, dass $\frac{p(p-1)}{2}$ Nicht-Null-Elemente in jeder der Matrizen $A^{(i)}$ für $1 \leq i \leq m$ enthalten sind, und einer zufälligen Zahl `rand_seed`.

n^2	m	rand_seed	nnz($\bar{A}$) (%)	nnz(L) (%)
1225	315	353153	1880 (0.487)	3673 (3.702)
2500	638	506383	3796 (0.238)	12432 (3.054)
4900	1243	7012433	7422 (0.122)	40046 (2.592)
10000	2525	10025253	15084 (0.058)	154908 (2.430)
19600	4935	14049353	29454 (0.030)	544214 (2.235)
40000	10050	200100503	60002 (0.015)	2327485 (2.304)
90000	22575	300225753	134816 (0.007)	11501478 (2.257)

Tabelle 3.1: Semidefinite Programme - Eingabedaten

Es wurde p=3 für jedes Testbeispiel gewählt. Die restlichen Eingaben für jedes Beispiel finden sich in Tabelle 3.1. In der Implementierung der APD- und HYBRID-Methode wird $\mathcal{A}$ durch eine Matrix $\bar{A}$ repräsentiert, die im Folgenden beschrieben wird:

Wir definieren $\mathsf{vec} : \mathcal{S}^n \to \mathbb{R}^{n^2}$ durch $\mathsf{vec}(X) := (X_{ij})_{\substack{i=1\dots n \\ j=1\dots n}}$. Wir setzen weiterhin

$$\bar{A} := \mathsf{vec}(\mathcal{A}) := \begin{bmatrix} \mathsf{vec}(A^{(1)})^T \\ \vdots \\ \mathsf{vec}(A^{(m)})^T \end{bmatrix}. \tag{3.9}$$

Wir geben in Tabelle 3.1 weiterhin die „Dichte" der Matrix $\bar{A}$ und des zugehörigen Cholesky Faktors L an, welcher für die HYBRID-Methode verwendet wurde – in anderen Worten die absolute Anzahl (und der relative Wert in %) der Nicht-Null-Elemente in der entsprechenden Matrix. Die Variable X wird als ein n^2-dimensionaler Vektor gespeichert, da $\bar{A}$ entsprechend n^2 Spalten enthält. Der Cholesky Faktor, der in SeDuMi berechnet wurde, hat in jedem Beispiel aus Tabelle 3.1 eine relative Dichte von etwa 50% Die Probleme in Tabelle 3.1 wurden mit SeDuMi und dem HYBRID- Verfahren berechnet. Kern der Untersuchung war die erreichbare Genauigkeit der Approximation für die Optimallösung, falls annehmbare Voraussetzungen erfüllt sind. Alle getesteten Probleme erfüllen Voraussetzung 3 und 4. Da SeDuMi die Optimallösung auf eine andere Weise als das HYBRID-Verfahren approximiert, wählen wir zum Vergleich ein Fehlermaß, welches sowohl die Unzulässigkeit bgl. **A** als auch bzgl. **K** bestraft.

Für ein gegebenes Tripel (X, y, S) definieren wir den primal-dualen Fehler als

$$error_{PD} := \left(\frac{\|X - \Pi_{\mathcal{S}^n_+}(X)\|_F^2}{1+\|X\|_F^2} + \frac{\|S - \Pi_{\mathcal{S}^n_+}(S)\|_F^2}{1+\|S\|_F^2} + \frac{\|XS\|_F^2}{1+\|X\|_F^2\|S\|_F^2} + \frac{\|\mathcal{A}(X)-b\|_2^2}{1+\|b\|_2^2} + \frac{\|\mathcal{A}^*(y)+S-C\|_F^2}{1+\|C\|_F^2} \right)^{\frac{1}{2}}.$$

$\mathtt{n}^2$	$\mathtt{m}$	$error_{PD}^{SeD}$	CPU-SeD	$error_{PD}^{HYB}$	CPU-HYB
1225	315	1.6040e-07	2.0	1.2854e-15	29.6
2500	638	1.3767e-07	5.5	1.1763e-15	65.6
4900	1243	1.6557e-07	14.2	1.7410e-15	117.6
10000	2525	1.0454e-07	91.5	1.4504e-15	375.5
19600	4935	5.1526e-08	610.0	1.2759e-14	995.1
40000	10050	3.1180e-08	5583.9	1.6705e-15	2844.3
90000	22575	7.3666e-08	52698.5	2.9179e-15	7496.8

Tabelle 3.2: Semidefinite Programme - Ergebnisse

In Tabelle 3.2 listen wir die Dimension der in Tabelle 3.1 beschriebenen Beispiele, die von SeDuMi erreichte Genauigkeit (wobei der Parameter pars.eps auf 0 gesetzt wurde, damit SeDuMi die bestmögliche Approximation errechnet) und die dafür benötigte Rechenzeit in Sekunden ($error_{PD}^{SeD}$, CPU-SeD) sowie die erreichte Genauigkeit und zugehörige Rechenzeit des HYBRID-Verfahrens ($error_{PD}^{HYB}$, CPU-HYB) auf. Bei Verwendung der Standardeinstellungen von SeDuMi betrug der abschließende Fehler $error_{PD}^{SeD}$ bei jedem Beispiel etwa $4 \cdot 10^{-6}$, wobei sich die dafür benötigte Rechenzeit im Vergleich zur Wahl von pars.eps=0 in etwa halbiert hat.
Für alle in Tabelle 3.1 gelisteten Beispiele liefert das HYBRID-Verfahren Approximationen von höherer Genauigkeit als SeDuMi. SeDuMi ist für kleine Dimensionen zwar deutlich schneller, bei den höheren Dimensionen ist HYBRID jedoch wesentlich effizienter. Abbildung 3.1 vermittelt einen Eindruck davon, in welcher Relation die Genauigkeit zur benötigten Rechenzeit bei beiden Verfahren steht.

Vergleich mit dem einfachen APD-Verfahren

In Tabelle 3.3 und 3.4 vergleichen wir das HYBRID-Verfahren mit dem in Abschnitt 2.6 beschriebenen APD-Verfahren. Berechnet wurden einige hochdimensionale semidefinite Programme mit einer Dimension von 400 oder mehr.

Die APD-Methode wurde ursprünglich zur Lösung von dünnbesetzten hochdimensionalen semidefiniten Programmen vorgeschlagen. Hauptziel der Methode war aber nicht die Berechnung von sehr genauen Lösungen sondern von „brauchbaren" Approximationen. Das Ziel des HYBRID-Verfahrens ist jedoch eine hohe Genauigkeit der Lösung.

Mit dem Zufallsgenerator [22] wurden dünnbesetzte semidefinite Programme mit Dimensionen zwischen 400 und 1000 sowie 30000 und mehr linearen Nebenbedingungen erzeugt. Die Probleme in Tabelle 3.3 wurden mit dem APD- und HYBRID-Verfahren berechnet. Die Abbruchbedingung für das APD-Verfahren

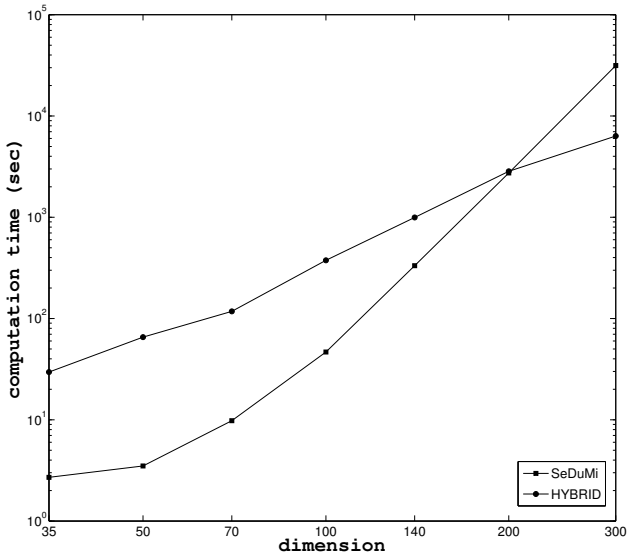

Abbildung 3.1: SeDuMi vs HYBRID, kleine Dimensionen

war die Berechnung von 4000 Iterationen, was bei Dimension 1000 mit dem TEST-PC fast zwei Tage in Anspruch nahm. Beim HYBRID-Verfahren wurde dagegen wie beim **Vergleich mit SeDuMi** solange gerechnet, bis die Optimallösung mit einer hohen Genauigkeit approximiert wurde ($\|\nabla\tilde{\phi}(X,S)\| + \|\nabla\tilde{f}(X,S)\| <$ `eps`). Da alle Iterierten der beiden Verfahren die linearen Bedingungen nach Konstruktion der Methoden erfüllen, wurde für diesen Vergleich das im Abschnitt **Kondition** (in 3.2.5) angegebene Fehlermaß verwendet.

In Tabelle 3.4 listen wir die Dimensionen der in Tabelle 3.3 beschriebenen Probleme, die berechnete Genauigkeit und benötigte Rechenzeit des APD-Verfahrens (error^{APD}, CPU-APD) sowie die entsprechenden Resultate der HYBRID-Methode (error^{HYB}, CPU-HYB) auf.

Die Ergebnisse in Tabelle 3.4 vermitteln den Eindruck, dass sich die APD- und die AHO-QMR-Methode gut ergänzen. Bei diesen Beispielen wurde beobachtet, dass die APD-Methode zu Beginn – weit weg von der Optimallösung – eine hohe Konvergenzrate besaß. Als die APD-Methode schließlich langsam wurde, befanden sich die Iterierten bereits in der „Schnelle Konvergenz"-Umgebung der AHO-QMR-Methode. Weiterhin verwendet der Präkonditionierer des in jedem AHO-QMR-Schritt zu lösenden Systems (T_4) den Cholesky Faktor von $\mathcal{A}\mathcal{A}^*$, welcher bereits zu Beginn der APD-Methode berechnet wurde.

Berechnung der Lovasz ϑ-Zahl

Das HYBRID-Verfahren wurde ebenfalls verwendet um die Lovasz-ϑ-Zahl von einigen zufälligen Graphen zu berechnen.

n²	m	rand_seed	nnz($\bar{A}$) (%)	nnz(L) (%)
160000	30000	4003030	179094 (0.004)	5602524 (0.620)
250000	30000	5003030	179062 (0.002)	179065 (0.020)
360000	40000	6004030	238830 (0.002)	148600 (0.009)
490000	50000	7005030	298404 (0.001)	112005 (0.004)
640000	70000	8007030	417856 (0.001)	245311 (0.005)
810000	100000	90010030	597130 (0.001)	1932642 (0.019)
1000000	100000	100010030	596964 (0.001)	222603 (0.002)

Tabelle 3.3: Hochdimensionale semidefinite Programme - Eingabedaten

n²	m	error^{APD}	CPU-APD	error^{HYB}	CPU-HYB
160000	30000	4.8026e-09	16688.9	4.6000e-14	7487.4
250000	30000	9.5350e-08	21923.3	6.2800e-13	8371.9
360000	40000	6.2031e-08	36430.7	1.0000e-15	16229.4
490000	50000	1.1035e-07	50053.0	2.4000e-14	29313.0
640000	70000	9.5654e-08	78572.9	0	32933.8
810000	100000	6.4942e-08	104463.2	1.2000e-14	68287.9
1000000	100000	1.0493e-07	142167.5	0	89141.6

Tabelle 3.4: Hochdimensionale semidefinite Programme - Ergebnisse

Für einen Graphen $G = (V, R)$, wobei V die Menge der Knoten und $R \subseteq V \times V$ die Menge der Kanten repräsentiert, ist das Problem der maximalen stabilen Menge folgendermaßen definiert:
Eine stabile Menge S in G ist eine Menge von Knoten, von denen jedes beliebige Paar nicht durch eine Kante verbunden ist. Die Größe einer stabilen Menge S entspricht ihrer Ordnung $|S|$. Die Stabilitätszahl $\alpha(G)$ des Graphen G ist die Größe einer stabilen Menge S, deren Ordnung maximal ist, d.h.

$$\alpha(G) = \max\{|S| \mid S \text{ ist stabile Menge von } G\}.$$

Die semidefinite Relaxierung dieses Problems ist

$$\vartheta(G) := \max\{e^T X e \mid I \bullet X = 1, X_{ij} = 0 \; \forall (i,j) \in R, i < j, \; X \succeq 0\},$$

wobei $e := (1, \ldots, 1)^T$. Eine ausführliche Besprechung des Problems findet sich z.B. in [11].
Sei $A^{(i,j)} \bullet X = 0$ die Repräsentierung der Bedingung $X_{ij} = 0$, wobei $A^{(i,j)}$ eine symmetrische $\{0, 1\}$-wertige Matrix sei. Es folgt dann, dass

$$A^{(i,j)} \bullet A^{(k,l)} = \begin{cases} 2, & falls \ (i,j) = (k,l) \\ 0, & falls \ (i,j) \neq (k,l) \end{cases}$$

und weiterhin

$$I \bullet I = n, \; I \bullet A^{(i,j)} = 0$$

gilt. Die Matrizen, welche die Gleichungsbedingungen beschreiben, sind daher paarweise orthogonal. Somit ist $\mathcal{A}\mathcal{A}^*$ eine Diagonalmatrix. Im Gegensatz dazu sind die zugehörigen in Innere-Punkte-Methoden verwendeten Systeme (welche eine Abbildung von der Form $\mathcal{A}\mathcal{F}\mathcal{A}^*$ mit einem linearen Operator $\mathcal{F}$ enthalten) normalerweise dicht besetzt. Daher sind Projektionsmethoden für die Berechnung der Lovasz ϑ-Zahl $\vartheta(G)$ geeignet, da die Projektion auf $\mathbf{A}$ hier sehr billig ist.
Die Ergebnisse der Berechnung von $\vartheta(G)$ werden in Tabelle 3.5 aufgelistet.

Wie in den vorherigen Abschnitten wird auch hier die quadrierte Dimension, die Anzahl der linearen Restriktionen, die Anzahl der Nicht-Null-Elemente in $\bar{A}$ sowie die Genauigkeit der vom HYBRID-Verfahren berechneten Approximation und die dafür benötigte Rechenzeit angegeben. Es wird dasselbe Fehlermaß wie im Abschnitt **Vergleich mit dem einfachen APD-Verfahren** verwendet. Der Cholesky Faktor L ist bei diesen Beispielen eine $\mathtt{m} \times \mathtt{m}$-Diagonalmatrix.

DIMACS Graphen

Abschließend betrachten wir einige Probleme der DIMACS-Sammlung [6]. Die zueghörigen Problemdaten und die Ergebnisse der Berechnung mit HYBRID werden in Tabelle 3.6 angegeben. Man erkennt, dass die Rechenzeit für *keller*5 die

$\mathtt{n}^2$	$\mathtt{m}$	$\mathtt{nnz}(\bar{A})$ (%)	error^{HYB}	CPU-HYB
40000	10011	20220 (0.0050)	0	2304.9
160000	39926	80250 (0.0013)	1.0000e-15	8489.6
490000	122624	245946 (0.0005)	0	29602.7
1000000	249670	500338 (0.0002)	0	87829.8

Tabelle 3.5: Lovasz ϑ-Zahl zufälliger Graphen - Problemdaten und Ergebnisse

Name	$\mathtt{n}^2$	$\mathtt{m}$	$\mathtt{nnz}(\bar{A})$ (%)	error^{HYB}	CPU-HYB
san_200_07_01	40000	5971	12140 (0.0051)	2.3100e-13	2415.0
brock400_1	160000	20078	40554 (0.0013)	1.8000e-14	4274.9
keller5	602176	74711	150196 (0.0003)	1.3000e-14	142882.6
brock800_1	640000	112096	224990 (0.0003)	2.0000e-15	21969.2

Tabelle 3.6: DIMACS Graphen - Problemdaten und Ergebnisse

mit Abstand längste ist: In diesem Fall hat HYBRID zuerst vom APD- zum AHO-QMR-Verfahren gewechselt und ist dann wieder zum APD-Verfahren zurückgekehrt, da die berechneten AHO-Schritte nicht „gut genug" waren. Dennoch wurde eine strikt komplementäre Optimallösung des Problems gefunden.

Fazit

Die in Abschnitt 3.2 vorgestellte Methode zur Lösung von hochdimensionalen semidefiniten Programmen verwendet nur teilweise Informationen zweiter Ordnung, liefert jedoch unter annehmbaren Voraussetzungen Approximationen von hoher Genauigkeit. Dies liegt zum Teil daran, dass die Kondition der Daten durch die beschriebenen Umformulierungen nicht verschlechtert wird.

Die zu Beginn geforderte Voraussetzung 3 wird bei der Umformulierung von System (3.5) zum System (T_4) nicht benötigt, sofern diese für einen Punkt (X^C, S^C) vorgenommen wird, welcher nach dem in Abschnitt 3.2.3 beschriebenen Verfahren erzeugt wurde. Daher ist nur Voraussetzung 1 (für die Existenz einer Optimallösung) und Voraussetzung 4 (für die Existenz des Cholesky Faktors) relevant, damit das HYBRID-Verfahren wohldefiniert ist. Ist sogar Voraussetzung 3 erfüllt, dann konvergiert das AHO-Verfahren lokal quadratisch. In der Nähe der Optimallösung können wir daher auch beim AHO-QMR-Verfahren eine entsprechende Konvergenzgeschwindigkeit erwarten.

Kapitel 4

Doppelt nichtnegative Programme

Wir werden uns in diesem Kapitel mit einer weiteren Art von konischen Programmen befassen, welche in den Raum $\mathcal{S}^n$ eingebettet sind. In vielen aktuellen Optimierungsproblemen ist außer dem Kegel $\mathcal{S}^n_+$ noch ein weiterer Kegel von besonderem Interesse: Es handelt sich um den vollständig positiven Kegel

$$\mathcal{C}^n_* := \{M^T M \mid M \in I\!R^{m\times n} \text{ für ein } m \in \mathbb{N},\ M \geq 0\}.$$

Formuliert werden die entsprechenden Probleme durch konische Programme mit $E = \mathcal{S}^n$ und $\mathcal{K} = \mathcal{C}^n_*$.

Es gilt sicherlich $\mathcal{C}^n_* \subseteq \mathcal{S}^n_+$. Der Kegel $\mathcal{C}^n_*$ ist jedoch problematisch. In [14] wurde gezeigt, dass die Beantwortung der Frage, ob ein Element zu diesem Kegel gehört oder nicht, NP-schwer ist. Ein Kegel, welcher als Approximation des vollständig positiven Kegels verwendet wird und deutlich einfacher zu handhaben ist, ist der doppelt nichtnegative Kegel

$$DNN^n := \{X \in \mathcal{S}^n \mid X \succeq 0, X \geq 0\} = \mathcal{S}^n_+ \cap \mathcal{N}^n,$$

wobei $\mathcal{N}^n := \{X \in \mathcal{S}^n \mid X \geq 0\}$ gelte. Es folgt dann $\mathcal{C}^n_* \subseteq DNN^n \subseteq \mathcal{S}^n_+$. Die zugehörigen konischen Programme werden im Folgenden definiert und die Verwendung der APD- und AHO-QMR-Methode zur Lösung solcher Probleme beschrieben.

4.1 Problemformulierung

Sei $E := \mathcal{S}^n$, $\mathcal{L} \subseteq E$ der Nullraum eines linearen Operators $\mathcal{A} : \mathcal{S}^n \to I\!R^m$ und $\bar{b} := \mathcal{A}(B)$ für ein $B \in E$. Falls $\mathcal{K} = DNN^n$, dann folgt $\mathcal{K}^D = \mathcal{S}^n_+ + \mathcal{N}^n$. Diese und weitere Eigenschaften des Kegels werden in [13] gezeigt. Aus dieser Quelle stammt der folgende Satz. Wie üblich bezeichnet hier $\mathcal{M}^\circ$ für eine Menge $\mathcal{M} \subseteq E$ das Innere der Menge $\mathcal{M}$.

Lemma 15. *Es gelten die folgenden Aussagen:*

i) $(\mathcal{S}^n_+)^\circ = \mathcal{S}^n_{++}$, $(\mathcal{N}^n)^\circ = \{X \in \mathcal{S}^n \mid X > 0\}$,

ii) DNN^n *ist ein abgeschlossener konvexer Kegel und es gilt* $(DNN^n)^\circ = \mathcal{S}^n_{++} \cap (\mathcal{N}^n)^\circ$,

iii) $(DNN^n)^D = \mathcal{S}^n_+ + \mathcal{N}^n$,

iv) $(DNN^n)^D$ *ist ein abgeschlossener konvexer Kegel und es gilt* $((DNN^n)^D)^\circ = \mathcal{S}^n_{++} + (\mathcal{N}^n)^\circ$.

Im Gegensatz zu semidefiniten Programmen ist der Kegel in diesem Fall nicht selbstdual.

Die Programme (P) und (D) lassen sich in der Form

$$(P_{DNN}) \qquad \min C \bullet X \mid \mathcal{A}(X) = \bar{b}, X \succeq 0, X \geq 0$$

und

$$(D_{DNN}) \quad \min B \bullet S \mid \exists y \in \mathbb{R}^m, S_1 \succeq 0, S_2 \geq 0 : \mathcal{A}^*(y) + S = C, S = S_1 + S_2,$$

schreiben, wobei y und S_1, S_2 „implizite Variablen" sind, die für die Darstellung von Elementen in $\mathcal{L}^\perp$ und $\mathcal{K}^D$ verwendet werden.

Die APD-Methode kann zur Lösung jedes beliebigen konischen Programmes verwendet werden, solange die Projektionen auf $\mathbf{A}$ und $\mathbf{K}$ leicht zu berechnen sind. Um dies für den Fall der doppelt nichnegativen Programme zu bewerkstelligen stellen wir die folgende Reformulierung vor:

$$(\tilde{P}) \quad \min \frac{1}{2}(C \bullet X + C \bullet X_{\mathcal{N}}) \mid \mathcal{A}(X) = \bar{b}, X - X_{\mathcal{N}} = 0, X \succeq 0, X_{\mathcal{N}} \geq 0$$

und

$$(\tilde{D}) \quad \min B \bullet S + B \bullet S_{\mathcal{N}} \mid \begin{array}{l} \exists y \in \mathbb{R}^m, Y \in \mathcal{S}^n : Y + \mathcal{A}^*(y) + S = \frac{1}{2}C, \\ S_{\mathcal{N}} - Y = \frac{1}{2}C, S \succeq 0, S_{\mathcal{N}} \geq 0. \end{array}$$

Hierbei ist $X_{\mathcal{N}}$ eine Schlupfvariable für das primale Problem und das Paar $(S, S_{\mathcal{N}})$ des dualen Programms entspricht dem impliziten Variablenpaar (S_1, S_2). In dieser Reformulierung hat der Kegel die Form $\mathcal{S}^n_+ \times \mathcal{N}^n$ und ist damit selbstdual. $(\tilde{D})$ erhält man als duales Programm zu $(\tilde{P})$; die Variable Y kann natürlich in $(\tilde{D})$ eliminiert werden.

Man kann leicht einsehen, dass die Probleme $(\tilde{P})$ und $(\tilde{D})$ zu (P_{DNN}) und (D_{DNN}) äquivalent sind. Inbesondere erfüllen $(\tilde{P})$ und $(\tilde{D})$ Voraussetzung 1 genau dann, wenn (P_{DNN}) und (D_{DNN}) Voraussetzung 1 erfüllen. Genauso verhält es sich bei Voraussetzung 2: $(\tilde{P})$ erfüllt die Slater-Bedingung genau dann, wenn (P_{DNN}) diese Bedingung erfüllt. Wegen $(DNN^n)^i = (DNN^n)^\circ$ folgt dies sofort aus Lemma

15 *ii*). Außerdem erfüllt $(\tilde{D})$ die Slater-Bedingung genau dann, wenn (D_{DNN}) sie erfüllt. Dies folgt mit $((DNN^n)^D)^i = ((DNN^n)^D)^\circ$ und Lemma 15 *iv*). Die Eindeutigkeit der Variable $\hat{S}$ in (D_{DNN}) impliziert jedoch nicht notwendigerweise die Eindeutigkeit von $(S, S_\mathcal{N})$, $\hat{S} = S + S_\mathcal{N}$, in $(\tilde{D})$.

Wenn $(\tilde{P})$, $(\tilde{D})$ die Slater-Bedingung erfüllen, dann sind ein für $(\tilde{P})$ zulässiger Punkt (X, X) und ein für $(\tilde{D})$ zulässiger Punkt $(S, S_\mathcal{N})$ beide optimal genau dann, wenn

$$C \bullet X + B \bullet S + B \bullet S_\mathcal{N} = B \bullet C \tag{4.1}$$

erfüllt ist. Wie in Abschnitt 1.1 bereits erwähnt wurde, ist (4.1) unter Voraussetzung 2 eine notwendinge aber in jedem Fall hinreichende Bedingung für eine Optimallösung.

Die Funktion $\tilde{\phi}$, welche in der APD-Methode verwendet wird, hat die Form

$$\begin{aligned}\tilde{\phi}((X, X_\mathcal{N}, S, S_\mathcal{N})) \;&= \tfrac{1}{2}(\|X - \Pi_{\mathcal{S}^n_+}(X)\|_F^2 + \|S - \Pi_{\mathcal{S}^n_+}(S)\|_F^2) \\ &\quad + \tfrac{1}{2}(\|X_\mathcal{N} - \Pi_{\mathcal{N}_n}(X_\mathcal{N})\|_F^2 + \|S_\mathcal{N} - \Pi_{\mathcal{N}^n}(S_\mathcal{N})\|_F^2)\end{aligned}$$

für $(X, X_\mathcal{N}, S, S_\mathcal{N},) \in \mathbf{A}$.
Hier sind der affine Raum $\mathbf{A}$ und der Kegel $\mathbf{K}$ Teilmengen des Raumes $\mathcal{S}^n \times \mathcal{S}^n \times \mathcal{S}^n \times \mathcal{S}^n$ und haben die folgende Form:

$$\begin{aligned}\mathbf{A} = \{(X, X_\mathcal{N}, S, S_\mathcal{N}) \mid\; & \mathcal{A}(X) = \bar{b}, X = X_\mathcal{N}, \\ & \exists y \in \mathbb{R}^m, Y \in \mathcal{S}^n : Y + \mathcal{A}^*(y) + S = \tfrac{1}{2}C, \\ & S_\mathcal{N} - Y = \tfrac{1}{2}C, \\ & C \bullet X + B \bullet S + B \bullet S_\mathcal{N} = B \bullet C\},\end{aligned}$$

$$\mathbf{K} = \;\mathcal{S}^n_+ \times \mathcal{N}^n \times \mathcal{S}^n_+ \times \mathcal{N}^n.$$

Eine sinnvolle Regularisierung ist durch Satz 11 für lineare und Semidefinitheits-Restriktionen gegeben. In den Abschnitten 4.2 und 4.3 geben wir eine darauf aufbauende Regularisierungsfunktion an und analysieren die Kosten einer Implementierung des APD-Verfahrens. Weiterhin werden Bedingungen betrachtet, für welche die Eindeutigkeit der Optimallösung bei der Reformulierung von (P_{DNN}) und (D_{DNN}) zu $(\tilde{P})$ und $(\tilde{D})$ erhalten bleibt. Es sei noch angemerkt, dass in [23] auch eine äquivalente Regularisierung für den Second-Order-Cone angegeben wurde.

4.2 Iterationskosten des Verfahrens

Es zeigt sich, dass jede Iteration der APD-Methode zur Lösung der Reformulierung $(\tilde{P})$ und $(\tilde{D})$ in etwa die gleichen Kosten erzeugt wie im semidefiniten Fall

$$\min C \bullet X \mid \mathcal{A}(X) = \bar{b}, X \succeq 0.$$

D.h.: Das Hinzufügen der Nichtnegativitätsbedingungen für die Komponenten von X erzeugt nur geringe zusätzliche Kosten.

Wie bereits beschrieben ist der teuerste Teil der APD-Methode für den semidefiniten Fall die einmalige Berechnung des Cholesky Faktors von $\mathcal{A}\mathcal{A}^*$, sowie die Projektionen auf $\mathbf{A}$ und $\mathbf{K}$. In dem Fall $(\tilde{P})$ und $(\tilde{D})$ liegen die Kosten der Projektion auf $\mathcal{N}^n$ in $\mathcal{O}(n^2)$ und sind daher zu vernachlässigen. Somit sind die Projektionen auf $\mathbf{K}$ praktisch genauso teuer wie im semidefiniten Fall.

Um die Kosten der Projektion auf $\mathbf{A}$ gering zu halten, definieren wir den linearen Operator $\bar{\mathcal{A}} : \mathcal{S}^n \times \mathcal{S}^n \to \mathbb{R}^{n^2} \times \mathbb{R}^m$ durch

$$\bar{\mathcal{A}}(X, X_{\mathcal{N}}) := \begin{bmatrix} -I & I \\ 0 & \mathsf{vec}(\mathcal{A}) \end{bmatrix} \mathsf{vec} \begin{pmatrix} X_{\mathcal{N}} \\ X \end{pmatrix}.$$

Für die Reformulierung $(\tilde{P})$ und $(\tilde{D})$ muss der Cholesky Faktor von $\bar{\mathcal{A}}\bar{\mathcal{A}}^*$ jedoch nicht explizit berechnet werden. Die Berechnung des Cholesky Faktors von $\mathcal{A}\mathcal{A}^*$ ist ausreichend: Ist $LL^T = \mathcal{A}\mathcal{A}^*$ gegeben, dann hat der Cholesky Faktor von $\bar{\mathcal{A}}\bar{\mathcal{A}}^*$ die Form

$$\bar{L} = \begin{bmatrix} \sqrt{2}I & 0 \\ \frac{1}{\sqrt{2}}\mathsf{vec}(\mathcal{A}) & \frac{1}{\sqrt{2}}L \end{bmatrix}.$$

Falls $(X, X_{\mathcal{N}}, S, S_{\mathcal{N}})$ eine Optimallösung von $(\tilde{P})$ und $(\tilde{D})$ ist, dann folgt $X \bullet S + X_{\mathcal{N}} \bullet S_{\mathcal{N}} = 0$ und somit $X \bullet S = 0$, $X_{\mathcal{N}} \bullet S_{\mathcal{N}} = 0$. Da $X \bullet S = 0 \Leftrightarrow XS = 0$ und $X_{\mathcal{N}} \bullet S_{\mathcal{N}} = 0 \Leftrightarrow X_{\mathcal{N}} \circ S_{\mathcal{N}} = 0$ gilt, definieren wir die Regularisierungsfunktion

$$\tilde{f}((X, X_{\mathcal{N}}, S, S_{\mathcal{N}})) := \|XS\|_F^2 + \|X_{\mathcal{N}} \circ S_{\mathcal{N}}\|_F^2.$$

Die Auswertungskosten der Ableitung von $\|X_{\mathcal{N}} \circ S_{\mathcal{N}}\|_F^2$ liegen in $\mathcal{O}(n^2)$ und sind damit ebenfalls vernachlässigbar.

Nach Satz 11 ist die Verwendung dieser Funktion als Regularisierung sinnvoll (man wähle $E := \mathcal{S}^n \times \mathcal{S}^1 \times \cdots \times \mathcal{S}^1$) und gewährleistet die positive Definitheit von $\partial^2(\tilde{\phi} + \tilde{f})(X^{opt}, X_{\mathcal{N}}^{opt}, S^{opt}, S_{\mathcal{N}}^{opt})$, sofern Voraussetzung 3 für $(\tilde{P})$ and $(\tilde{D})$ erfüllt ist.

Wir zeigen als Nächstes wie Voraussetzung 3 mit den Ausgangsproblemen (P_{DNN}) and (D_{DNN}) zusammenhängt.

4.3 Regularität der selbstualen Reformulierung

Für konische Programme, deren Kegel nicht notwendigerweise selbstdual sind, verwenden wir die folgende naheliegende Verallgemeinerung der strikten Komplementarität für einen belieibigen nicht-leeren, abgeschlossenen und konvexen Kegel $\mathcal{K}$ im zugehörigen euklidischen Raum E:

Definition 9. *Die Optimallösung (x^{opt}, s^{opt}) eines Paares von konischen primal-dualen Programmen (P) und (D) heißt strikt komplementär, falls*

$$x^{opt} \in \left(\mathcal{K} \cap \{\ h \mid \langle h, s^{opt}\rangle = 0\}\right)^i \tag{4.2}$$

und

$$s^{opt} \in \left(\mathcal{K}^D \cap \{\ h \mid \langle h, x^{opt}\rangle = 0\}\right)^i. \tag{4.3}$$

Falls $\mathcal{K} = \mathcal{K}^D$ der lineare oder der semidefinite Kegel ist, d.h. $\mathcal{K} = I\!R^n_+$ oder $\mathcal{K} = \mathcal{S}^n_+$, dann prüft man leicht nach, dass Bedingung (4.2) zur Bedingung (4.3) oder zur Bedingung "$x^{opt} + s^{opt} \in \mathcal{K}^\circ$" äquivalent ist.

Für den linearen oder semidefiniten Kegel sind die Bedingungen (4.2) oder (4.3) auch äquivalent zu jeder der folgenden Bedingungen:

$$\mathsf{aff}(\mathcal{NC}(\mathcal{K}, x^{opt})) \subseteq \mathsf{aff}(\mathcal{NC}(\mathcal{K}^D, s^{opt}))^\perp \tag{4.4}$$

oder

$$\{\ h \in \mathcal{K}\ \mid\ \langle h, s^{opt}\rangle = 0\}\ \cap\ \{\ h \in \mathcal{K}^D\ \mid\ \langle h, x^{opt}\rangle = 0\}\ =\ \{0\}. \tag{4.5}$$

In Bedingung (4.4) steht $\mathcal{NC}$ für den Normalenkegel. Dieser ist für eine konvexe Menge $\mathcal{M} \subseteq E$ und einen Punkt $x \in \mathcal{M}$ folgendermaßen definiert:

$$\mathcal{NC}(\mathcal{M}, x) := \{z \in E \mid \langle z, y - x\rangle_E \leq 0\ \forall y \in \mathcal{M}\}.$$

Wir werden als Nächstes sehen, dass (4.2) für allgemeinere Kegel nicht zu (4.3) äquivalent ist und dass es konische Programme gibt, die zwar Bedingung (4.5) aber nicht (4.4) erfüllen, und dass es auch Probleme gibt, welche Bedingung (4.4) aber nicht (4.2) oder (4.3) erfüllen.

Die Relation der Bedingungen (4.2) – (4.5), welche im LP- oder SDP-Fall alle äquivalent sind, kann folgendermaßen zusammengefasst werden:

$$\{(4.2),\ (4.3)\} \underset{\not\Longleftarrow}{\Longrightarrow} (4.4) \underset{\not\Longleftarrow}{\Longrightarrow} (4.5), \qquad (4.2) \underset{\not\Longleftarrow}{\not\Longrightarrow} (4.3).$$

Repräsentation des Normalenkegels

Sei $x \in \mathcal{NC}(\mathcal{K}^D, s^{opt})$, d.h. $\langle x, k^D - s^{opt}\rangle \leq 0$ für alle $k^D \in K^D$. Die Wahl von $k^D = 0$ impliziert $\langle x, s^{opt}\rangle \geq 0$. Wegen $k^D \in \mathcal{K}^D$ folgt $\lambda k^D \in \mathcal{K}^D$ für alle $\lambda > 0$ und somit $\langle x, k^D\rangle \leq 0$ für alle $k^D \in \mathcal{K}^D$, d.h. $-x \in \mathcal{K}$. Darsus folgt $\langle x, s^{opt}\rangle \leq 0$ und folglich

$$\langle x, s^{opt}\rangle = 0.$$

Somit gilt $-x \in \mathcal{K} \cap \{\ h \mid \langle h, s^{opt}\rangle = 0\}$ – die rechte Schnittmenge in Bedingung (4.5). Ist andererseits $x \in -\mathcal{K} \cap \{\ h \mid \langle h, s^{opt}\rangle = 0\}$ gegeben, dann folgt $\langle x, k^D - s^{opt}\rangle \leq 0$ für alle $k^D \in \mathcal{K}^D$, womit wir

$$\mathcal{NC}(\mathcal{K}^D, s^{opt}) = -\mathcal{K} \cap \{\ h \mid \langle h, s^{opt}\rangle = 0\} \tag{4.6}$$

erhalten. Analog folgt

$$\mathcal{NC}(\mathcal{K}, x^{opt}) = -\mathcal{K}^D \cap \{\ h \mid \langle h, x^{opt}\rangle = 0\}. \tag{4.7}$$

Zunächst zeigen wir ein Beispiel dafür, dass die Eindeutigkeit einer Optimallösung für (P_{DNN}) und (D_{DNN}), welche Bedingung (4.5) oder (4.4) erfüllt, nicht notwendigerweise die Eindeutigkeit oder strikte Komplementarität einer äquivalenten Lösung in der selbstdualen Formulierung $(\tilde{P})$ und $(\tilde{D})$ impliziert:

Beispiel 4. *Betrachte das primal-duale Paar*

$$(P_{DNN}) \qquad \min \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \bullet X \ \mid \ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \bullet X = 1, X \succeq 0, X \geq 0$$

und

$$(D_{DNN}) \qquad \max y \ \mid \ \begin{pmatrix} y & 0 \\ 0 & y \end{pmatrix} + S_1 + S_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, S_1 \succeq 0, S_2 \geq 0.$$

Definiere $\bar{X} := \begin{pmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} \end{pmatrix}$. *Dann gilt* $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \bullet \bar{X} = 1$, $\bar{X} \succ 0$ *und* $\bar{X} > 0$. *Somit erfüllt* (P_{DNN}) *die Slater-Bedingung.*

Für $\bar{y} := -\frac{1}{4}$, $\bar{S}_1 := \begin{pmatrix} \frac{1}{2} & -\frac{1}{8} \\ -\frac{1}{8} & \frac{1}{8} \end{pmatrix} \succ 0$, $S_2 := \begin{pmatrix} \frac{3}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} \end{pmatrix} > 0$ *gilt*

$$\begin{pmatrix} \bar{y} & 0 \\ 0 & \bar{y} \end{pmatrix} + \bar{S}_1 + \bar{S}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Damit erfüllt auch (D_{DNN}) *die Slater-Bedingung.*

Man prüft leicht nach, dass

$$X^{opt} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \ S^{opt} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

die eindeutige primal-duale Optimallösung ist. Es gilt weiterhin

$$\mathcal{K} \cap \{X \mid X \bullet S^{opt} = 0\} = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & \mu \end{pmatrix} \mid \mu \geq 0 \right\}$$

und

$$\mathcal{K}^D \cap \{S \mid S \bullet X^{opt} = 0\} = \left\{ \begin{pmatrix} \mu & \nu \\ \nu & 0 \end{pmatrix} \mid \mu, \nu \geq 0 \right\}.$$

Somit erfüllen X^{opt}, S^{opt} zwar Bedingung (4.2) aber nicht (4.3). Mit Hilfe von (4.6) und (4.7) sieht man ebenfalls schnell ein, dass die Optimallösung (4.4) und (4.5) erfüllt.

Definiere $S(\lambda) := \begin{pmatrix} \lambda & 0 \\ 0 & 0 \end{pmatrix}$ für $\lambda \in [0,1]$. Dann gilt $S(\lambda) \succeq 0$, $S(\lambda) \geq 0$ und $S(\lambda) + S(1-\lambda) = S^{opt}$. Obwohl X^{opt} und S^{opt} die eindeutige Optimallösung von (P_{DNN}) und (D_{DNN}) sind, ist die Zerlegung von S^{opt} in Elemente aus $\mathcal{S}^2_+$ und $\mathcal{N}^2$ nicht eindeutig. Weiterhin ist jede Zerlegung von S^{opt}, die zusammen mit (X^{opt}, X^{opt}) eine Optimallösung von $(\tilde{P})$ und $(\tilde{D})$ bildet, nicht strikt komplementär.

Sei $X \in DNN = \mathcal{S}^n_+ \cap \mathcal{N}^n$ gegeben. Sei weiterhin $X = UDU^T$ die Eigenwertzerlegung von X mit einer orthogonalen Matrix $U = (u_1, \ldots, u_n)$ und einer Diagonalmatrix D mit $D_{11} \geq D_{22} \geq \ldots \geq D_{nn}$. Sei k die Anzahl der positiven Diagonaleinträge von D und $R := \{(i,j) \mid X_{ij} = 0,\ i \leq j\}$ die Menge der aktiven Nichtnegativitätsbedingungen. Wir setzen

$$\begin{aligned} T_1(X) &:= \mathsf{aff}(\mathcal{NC}(\mathcal{S}^n_+, X)), \\ T_2(X) &:= \mathsf{aff}(\mathcal{NC}(\mathcal{N}^n, X)). \end{aligned}$$

In diesem Fall gilt

$$T_1(X) = \{ \sum_{\substack{i,j=k+1 \\ i \leq j}}^{n} \mu_{i,j}(u_i u_j^T + u_j u_i^T) \quad | \quad \mu_{i,j} \in \mathbb{R},\ k+1 \leq i \leq j \leq n \}$$

und

$$T_2(X) = \{ \sum_{r \in R} \nu_r E_r \quad | \quad \nu_r \in \mathbb{R},\ r \in R \},$$

wobei $E_{(i,j)}$ die Matrix mit $[E_{(i,j)}]_{ij} = [E_{(i,j)}]_{ji} = 1$ und $[E_{(i,j)}]_{pq} = 0$ für $\{p,q\} \neq \{i,j\}$ bezeichnet.

Lemma 16. *Sei $X \in \mathcal{S}^n_+ \cap \mathcal{N}^n$ und $S \in \mathcal{S}^n_+ + \mathcal{N}^n$ mit $X \bullet S = 0$ gegeben.*
Seien $S_1 \in \mathcal{S}^n_+$, $S_2 \in \mathcal{N}^n$ gegeben, welche $S_1 + S_2 = S$ erfüllen.

Falls $T_1(X) \cap T_2(X) = \{0\}$ gilt, dann sind S_1 und S_2 eindeutig.

Beweis. Angenommen, es gilt $S = S_1 + S_2 = W_1 + W_2$ für $S_1, W_1 \in \mathcal{S}^n_+$, $S_2, W_2 \in \mathcal{N}^n$. Aus $X \in \mathcal{S}^n_+ \cap \mathcal{N}^n$ und $X \bullet S = 0$ folgt $X \bullet S_1 = X \bullet W_1 = X \bullet S_2 = X \bullet W_2 = 0$ und somit $XS_1 = XW_1 = X \circ S_2 = X \circ W_2 = 0$.
Sei $Y_1 := \frac{1}{2}(S_1 + W_1) \in \mathcal{S}^n_+$ und $Y_2 := \frac{1}{2}(S_2 + W_2) \in \mathcal{N}^n$. Dann gilt $S = Y_1 + Y_2$. Für $H := S_1 - W_1 = W_2 - S_2$ erhalten wir $XH = 0$ und $X \circ H = 0$. Somit folgt $H \in T_1(X) \cap T_2(X)$. Dies impliziert $H = 0$, d.h. $S_1 = W_1$ and $S_2 = W_2$ und damit folgt die Behauptung. □

Ein nichttriviales komplementäres Paar (X, S), welches eine eindeutige Zerlegung $S \in \mathcal{S}^n_+ + \mathcal{N}^n$ besitzt, wird im folgenden Beispiel angegeben:

Beispiel 5. *Setze*

$$X := \begin{pmatrix} 1 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \end{pmatrix} \in \mathcal{S}^3_+ \cap \mathcal{N}^3.$$

Für

$$U := \begin{pmatrix} \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}, \; D := \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

gilt $X = UDU^T$.
Es gilt weiterhin

$$T_1(X) = \{ \; \mu \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & -\frac{1}{2\sqrt{2}} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2} \end{pmatrix} \quad | \quad \mu \in \mathbb{R} \; \}$$

und

$$T_2(X) = \{ \; \nu \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad | \quad \nu \in \mathbb{R} \; \}.$$

Man sieht leicht ein, dass $T_1(X) \cap T_2(X) = \{0\}$ *gilt.*
Sei $S \in \mathcal{S}^3_+ + \mathcal{N}^3$ *so gegeben, dass* $X \bullet S = 0$ *gilt. Die Eindeutigkeit der Zerlegung* $S = S_1 + S_2$ *folgt direkt aus Lemma 16.*

Satz 12. *Falls beide Bedingungen (4.2) und (4.3) erfüllt sind, dann ist auch Bedingung (4.4) erfüllt. Weiterhin impliziert (4.4) Bedingung (4.5).*
Ist $\mathcal{K}$ *der doppelt nichtnegative Kegel,* (X^{opt}, S^{opt}) *die eindeutige Optimallösung eines primal-dualen Paares* (P_{DNN}) *und* (D_{DNN})*, welche Bedingung (4.3) erfüllt, und gilt* $T_1(X^{opt}) \cap T_2(X^{opt}) = \{0\}$ *(siehe Lemma 16), dann besitzt auch die Reformulierung* $(\tilde{P})$, $(\tilde{D})$ *eine eindeutige und strikt komplementäre Optimallösung.*

Beweis. Wir zeigen zuerst, dass (4.2) und (4.3) zusammen Bedingung (4.4) implizieren:

Seien (4.2) und (4.3) erfüllt. Dann existiert ein $\varepsilon > 0$, so dass

$$(s^{opt} + \Delta s) \in \mathcal{K}^D \; \forall \Delta s \text{ mit } \|\Delta s\|_2 = \varepsilon, \Delta s \in \mathsf{aff}\mathcal{NC}(\mathcal{K}, x^{opt})$$

und

$$(x^{opt} + \Delta x) \in \mathcal{K} \; \forall \Delta x \text{ mit } \|\Delta x\|_2 = \varepsilon, \Delta x \in \mathsf{aff}\mathcal{NC}(\mathcal{K}^D, s^{opt}).$$

Weiterhin gilt

$$\{-s^{opt} + \lambda\Delta s \;\mid\; \lambda \in \mathbb{R},\ \|\Delta s\|_2 = \varepsilon, \Delta s \in \mathsf{aff}\mathcal{NC}(\mathcal{K}, x^{opt})\} = \mathsf{aff}\mathcal{NC}(\mathcal{K}, x^{opt})$$

und

$$\{-x^{opt} + \mu\Delta x \;\mid\; \mu \in \mathbb{R},\ \|\Delta x\|_2 = \varepsilon, \Delta x \in \mathsf{aff}\mathcal{NC}(\mathcal{K}^D, s^{opt})\} = \mathsf{aff}\mathcal{NC}(\mathcal{K}^D, s^{opt}).$$

Sei ein beliebiges Element aus $\mathsf{aff}\mathcal{NC}(\mathcal{K}, x^{opt})$ in der Form $-s^{opt} + \lambda\Delta s$ mit $\|\Delta s\|_2 = \varepsilon$ gegeben. Sei weiterhin ein Element aus $\mathsf{aff}\mathcal{NC}(\mathcal{K}^D, s^{opt})$ durch $-x^{opt} + \mu\Delta x$ mit $\|\Delta x\|_2 = \varepsilon$ gegeben. Dann folgt

$$\begin{aligned}\langle -x^{opt} + \mu\Delta x, -s^{opt} + \lambda\Delta s\rangle = \;& \langle -x^{opt}, -s^{opt}\rangle + \langle -x^{opt}, \lambda\Delta s\rangle \\ & + \langle -s^{opt}, \mu\Delta x\rangle + \lambda\mu\langle \Delta x, \Delta s\rangle \\ = \;& \lambda\mu\langle \Delta x, \Delta s\rangle.\end{aligned}$$

Wegen

$$\langle \underbrace{s^{opt} \pm \Delta s}_{\in\mathcal{K}^D}, \underbrace{x^{opt} \pm \Delta x}_{\in\mathcal{K}}\rangle = \langle \pm\Delta s, \pm\Delta x\rangle \geq 0$$

folgt $\langle \Delta s, \Delta x\rangle = 0$ und somit

$$\langle -x^{opt} + \mu\Delta x, -s^{opt} + \lambda\Delta s\rangle = 0.$$

Dies impliziert $\mathsf{aff}\mathcal{NC}(\mathcal{K}, x^{opt}) \subseteq \mathsf{aff}\mathcal{NC}(\mathcal{K}^D, s^{opt})^\perp$, d.h. Bedingung (4.4) ist erfüllt.

Weiterhin folgt direkt aus (4.6) und (4.7), dass (4.4) Bedingung (4.5) impliziert.

Von nun an gelte $E = \mathcal{S}^n$ und $\mathcal{K} = DNN^n$. Für $X^{opt} \in DNN^n$ und $S^{opt} \in (DNN^n)^D$ sei Bedingung (4.3) und $T_1(X^{opt}) \cap T_2(X^{opt}) = \{0\}$ erfüllt. Sei weiterhin $S^{opt} = S^1 + S^2$, $S^1 \in \mathcal{S}^n_+$, $S^2 \in \mathcal{N}^n$, die eindeutige Zerlegung von S^{opt} in Elemente aus $\mathcal{S}^n_+$ und $\mathcal{N}^n$.
Es gilt sicherlich $X^{opt} + S^1 \succeq 0$ und $X^{opt} + S^2 \geq 0$.
Angenommen, es gilt $X^{opt} + S^1 \not\succ 0$. Wegen $X^{opt}S^1 = 0$ existiert ein $\Delta S^1 \in \mathcal{S}^n_+$, $\Delta S^1 \neq 0$, so dass $X^{opt}\Delta S^1 = S^1\Delta S^1 = 0$ gilt. Angenommen, $S^{opt} - \varepsilon\Delta S^1 \in \mathcal{K}^D$ für ein festes $\varepsilon > 0$. Dann existieren $\tilde{S}^1 \in \mathcal{S}^n_+$ und $\tilde{S}^2 \in \mathcal{N}^n$ mit $\tilde{S}^1 + \tilde{S}^2 = S^{opt} - \varepsilon\Delta S^1$. Daraus folgt $(\tilde{S}_1 + \varepsilon\Delta S^1) + \tilde{S}^2 = S^{opt}$. Wegen $S^1 - \varepsilon\Delta S^1 \notin \mathcal{S}^n_+$ folgt $\tilde{S}^1 + \varepsilon\Delta S^1 \neq S^1$, was der Eindeutigkeit der Zerlegung widerspricht. Es folgt somit, dass $S^{opt} - \varepsilon\Delta S^1 \notin \mathcal{K}^D$ für alle $\varepsilon > 0$ gilt. Wegen $S^{opt} + \varepsilon\Delta S^1 \in \mathcal{K}^D \cap \{Z \mid Z \bullet X^{opt} = 0\}$ für alle $\varepsilon > 0$ ist dies ein Widerspruch zu (4.3).
Als Nächstes nehmen wir an, dass $X^{opt} + S^2 \not\succeq 0$. Dann existiert ein $\Delta S^2 \in \mathcal{N}^n$, $\Delta S^2 \neq 0$, so dass $X^{opt} \circ \Delta S^2 = S^2 \circ \Delta S^2 = 0$ erfüllt ist. Angenommen, $S^{opt} - \varepsilon\Delta S^2 \in \mathcal{K}^D$ für ein $\varepsilon > 0$. Dann existieren $\hat{S}^1 \in \mathcal{S}^n_+$ und $\hat{S}^2 \in \mathcal{N}^n$ mit $\hat{S}^1 + \hat{S}^2 = S^{opt} - \varepsilon\Delta S^2$, womit sich $\hat{S}^1 + (\hat{S}^2 + \varepsilon\Delta S^2) = S^{opt}$ ergibt. Wegen $S^2 - \varepsilon\Delta S^2 \notin \mathcal{N}^n$ folgt, dass $\hat{S}^2 + \varepsilon\Delta S^2 \neq S_2$ gilt – dies ist ein Widerspruch zur Eindeutigkeit der Zerlegung. Es folgt $S^{opt} - \varepsilon\Delta S^2 \notin \mathcal{K}^D$ für alle $\varepsilon > 0$. Wie oben zeigt man, dass dies ein Widerspruch zu (4.3) ist. □

Abschließend zeigen wir, dass die Bedingungen (4.4) und (4.5) i.Allg. nicht äquivalent sind.

Beispiel 6. *Setze*

$$X := \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \in DNN^3, \quad S := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \in (DNN^3)^D.$$

Es gilt $X \bullet S = 0$. *Für*

$$U := \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{pmatrix}$$

folgt

$$X = U \begin{pmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} U^T, \; S = U \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} U^T.$$

Sei $-(H_1 + H_2) \in \mathcal{NC}(DNN^3, X)$, $H_1 \in \mathcal{S}^3_+$, $H_2 \in \mathcal{N}^3$ *gegeben. Dann gilt* $(H_1 + H_2) \bullet X = 0$ *genau dann, wenn* $H_1 X = 0$ *und* $H_2 \circ X = 0$ *gilt. Wegen* $X > 0$ *ist dies äquivalent zu* $H_2 = 0$ *und*

$$H_1 = \alpha \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} + \beta \begin{pmatrix} 0 & 2 & -2 \\ 2 & -2 & 0 \\ -2 & 0 & 2 \end{pmatrix} + \gamma \begin{pmatrix} 4 & -2 & -2 \\ -2 & 1 & 1 \\ -2 & 1 & 1 \end{pmatrix}$$

für entsprechende Elemente $\alpha, \beta, \gamma \in \mathbb{R}$, *so dass* $H_1 \succeq 0$ *gilt. Es folgt somit*

$$H_1 = U \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2\alpha & \sqrt{12}\beta \\ 0 & \sqrt{12}\beta & 6\gamma \end{pmatrix} U^T.$$

Die Wahl von $(\alpha, \beta, \gamma) \in \{(1,0,0),\ (1,0,1),\ (6,1,2)\}$ *erzeugt drei linear unabhängige, positiv semidefinite Matrizen* H_1. *Daraus folgt*

$$\mathsf{aff}(\mathcal{NC}(DNN^3, X)) = \{U \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2\alpha & \sqrt{12}\beta \\ 0 & \sqrt{12}\beta & 6\gamma \end{pmatrix} U^T \;\mid\; \alpha, \beta, \gamma \in \mathbb{R}\}.$$

Sei nun $-H \in \mathcal{NC}((DNN^3)^D, S)$, $H \in DNN^3$, *gegeben. Wegen* $S \succeq 0$ *folgt* $H \bullet S = 0 \Leftrightarrow HS = 0$ *und damit*

$$\begin{aligned} H &= \lambda \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} + \mu \begin{pmatrix} -4 & -1 & -1 \\ -1 & 2 & 2 \\ -1 & 2 & 2 \end{pmatrix} + \nu \begin{pmatrix} 4 & -2 & -2 \\ -2 & 1 & 1 \\ -2 & 1 & 1 \end{pmatrix} \\ &= U \begin{pmatrix} 3\lambda & 0 & \sqrt{18}\mu \\ 0 & 0 & 0 \\ \sqrt{18}\mu & 0 & 6\nu \end{pmatrix} U^T, \end{aligned}$$

so dass $H \succeq 0$ *und* $H \geq 0$ *erfüllt ist.*
Für $(\lambda, \mu, \nu) \in \{(1,0,0),\ (3,0,1),\ (6,1,2)\}$ *erhalten wir entsprechende linear unabhängige* $H \in DNN^3$*. Daher folgt*

$$\mathsf{aff}(\mathcal{NC}((DNN^3)^D, S)) = \{U \begin{pmatrix} 3\lambda & 0 & \sqrt{18}\mu \\ 0 & 0 & 0 \\ \sqrt{18}\mu & 0 & 6\nu \end{pmatrix} U^T \ \mid \ \lambda, \mu, \nu \in \mathbb{R}\}.$$

Es folgt $\mathsf{aff}(\mathcal{NC}(DNN^3, X)) \nsubseteq \mathsf{aff}(\mathcal{NC}((DNN^3)^D, S))^\perp$*, d.h.: Dieses Beispiel erfüllt Bedingung (4.4) nicht.*

Die Äquivalenz

$$\begin{aligned} & H \in DNN^3 \cap \{H \mid H \bullet S = 0\} \ \cap \ (DNN^3)^D \cap \{H \mid H \bullet X = 0\} \\ \iff & -H \in \mathcal{NC}(DNN^3, X) \cap \mathcal{NC}((DNN^3)^D, S) \\ \iff & H = U \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 6\gamma \end{pmatrix} U^T = \gamma \begin{pmatrix} 4 & -2 & -2 \\ -2 & 1 & 1 \\ -2 & 1 & 1 \end{pmatrix}, \ H \succeq 0, H \geq 0, \end{aligned}$$

impliziert $\gamma = 0$*. Daher ist Bedingung (4.5) erfüllt.*

Ist für ein primal-duales Paar (P_{DNN}) und (D_{DNN}) die Optimallösung (X^{opt}, S^{opt}) eindeutig, gilt weiterhin Bedingung (4.3) und $T_1(X^{opt}) \cap T_2(X^{opt}) = \{0\}$, dann folgt nach Satz 12 und Lemma 16, dass auch die Reformulierung $(\tilde{P})$ und $(\tilde{D})$ eine eindeutige strikt komplentäre Optimallösung besitzt, womit die Voraussetzungen von Satz 11 erfüllt sind. Somit folgt $\partial^2(\tilde{\phi} + \tilde{f})(X^{opt}, X^{opt}, S_1^{opt}, S_2^{opt}) \succ 0$. In diesem Fall kann bei Verwendung der APD-Methode zur Lösung der Reformulierung eine bessere Konvergenz erwartet werden.

4.4 Numerische Ergebnisse

Die APD-Methode ist verwendbar um Probleme der Form (P_{DNN}) und (D_{DNN}) mit Hilfe der Reformulierung $(\tilde{P})$ und $(\tilde{D})$ zu lösen. Die hier genutzte MATLAB® Implementierung entspricht dabei derjenigen, die in Abschnitt 2.6 bereits beschrieben wurde. Die zusätzlichen für den *DNN*-Kegel erforderlichen Berechnungen wurden entsprechend Abschnitt 4.2 implementiert. Weiterhin lässt sich die in Abschnitt 3.2 beschriebene HYBRID-Methode erweitern, um die Probleme $(\tilde{P})$ und $(\tilde{D})$ zu lösen. Die Arbeitsweise der erweiterten Version wird in Abschnitt 4.5 kurz beschrieben. Sie wurde zur Lösung der folgenden Probleme verwendet:

Problem der maximalen stabilen Menge

Es wurden einige zufällige Graphen der Dimension `n` mit einer Adjazenzmatrix A_G, welche etwa zur Hälfte besetzt ist, erstellt. Die doppelt nichtnegative Relaxierung des Problems der maximalen stabilen Menge (siehe Abschnitt 3.2.6) für

diese Graphen kann in der Form

$$(P_{DNN}) \quad \max\{e^T X e \mid I \bullet X = 1, X_{ij} = 0 \ \forall (i,j) \in R, i < j, \ X \succeq 0, \ X \geq 0\},$$

$e := (1, \ldots, 1)^T$, formuliert werden.

Tabelle 4.1 enthält alle relevanten Problemdaten. Sie werden im Folgenden kurz beschrieben:
In der Implementierung wird der lineare Operator $\mathcal{A}$ durch die Matrix $\bar{A}$ mit `m` Zeilen und $\mathtt{n}^2$ Spalten beschrieben (siehe (3.9)), so dass die Variable X als $\mathtt{n}^2$-dimensionaler Vektor gespeichert wird. Die „Dünnbesetztheit" der Matrix $\bar{A}$ wird durch die absolute (und relative Anzahl in %) von Nicht-Null-Einträgen in der Spalte $\mathtt{nnz}(\bar{A})$ angegeben. Der Cholesky Faktor L von $\bar{A}\bar{A}^T$ ist eine Diagonalmatrix, wie in Abschnitt 3.2.6 bereits erläutert wurde. Für jedes Beispiel wird außerdem die Optimallösung der semidefiniten bzw. doppelt nichtnegativen Relaxierung in der Spalte `SDP OPT` bzw. `DNN OPT` angegeben. Diese Werte wurden mit der in Abschnitt 3.2 vorgestellten HYBRID-Methode berechnet.

$\mathtt{n}^2$	`m`	$\mathtt{nnz}(A)$ (%)	`SDP OPT`	`DNN OPT`
2500	638	1324 (0.0830)	7.311	7.269
10000	2381	4860 (0.0204)	10.917	10.845
40000	10011	20220 (0.0050)	14.424	14.328
160000	39926	80250 (0.0013)	20.279	20.138
490000	122624	245946 (0.0005)	26.561	26.386
1000000	249670	500338 (0.0002)	31.823	31.612

Tabelle 4.1: Doppelt nichtnegative „maximale stabile Menge" Relaxierung

Die Reformulierung $(\tilde{P})$ und $(\tilde{D})$ kann sowohl für HYBRID als auch für SeDuMi und das ebenfalls zur Lösung von konischen Programmen gedachte Softwarepaket SDPNAL (siehe [26]) genutzt werden. Die Verwendung von SeDuMi für Probleme mit höheren Dimensionen war auf unserem TEST-PC allerdings nicht möglich. Für $\mathtt{n} = 200$ wurde SeDuMi nach zehn Stunden gestoppt; zu der Zeit hatte es noch nicht einmal die erste Iteration beendet.

Zu Vergleichszwecken definieren wir den primal-dualen Fehler durch

$$\begin{aligned} error_{PD} := \Bigg(& \frac{\|X - \Pi_{\mathcal{S}^n_+}(X)\|_F^2 + \|X - \Pi_{\mathcal{N}^n}(X)\|_F^2}{1 + \|X\|_F^2} \\ & + \frac{\|S - \Pi_{\mathcal{S}^n_+}(S)\|_F^2}{1 + \|S\|_F^2} + \frac{\|S_{\mathcal{N}} - \Pi_{\mathcal{N}^n}(S_{\mathcal{N}})\|_F^2}{1 + \|S_{\mathcal{N}}\|_F^2} \\ & + \frac{\|XS\|_F^2}{1 + \|X\|_F^2 \|S\|_F^2} + \frac{\|X \circ S_{\mathcal{N}}\|_F^2}{1 + \|X\|_F^2 \|S_{\mathcal{N}}\|_F^2} \\ & + \frac{\|\mathcal{A}(X) - b\|_2^2}{1 + \|b\|_2^2} + \frac{\|\mathcal{A}^*(\bar{y}) + (S + S_{\mathcal{N}}) - C\|_F^2}{1 + \|C\|_F^2} \Bigg)^{\frac{1}{2}}. \end{aligned}$$

In Tabelle 4.2 listen wir die Dimensionen der in Tabelle 4.1 beschriebenen Probleme auf. Der finale von SeDuMi ermittelte primal-duale Fehler und die zugehörige Rechenzeit (in Sekunden) werden als ($error^S_{PD}$, `CPU-S`) gelistet. Die Spalten ($error^N_{PD}$, `CPU-N`) bzw. ($error^H_{PD}$, `CPU-H`) beinhalten die entsprechenden von SDPNAL bzw. HYBRID erzeugten Daten. Das Ziel von HYBRID ist die Berechnung von Lösungen hoher Genauigkeit. Für einen geeigneten Vergleich wurden daher auch SeDuMi und SDPNAL so eingestellt, dass sie die bestmöglichen Approximationen der Optimallösung berechnen. Die optimalen Werte, die für die doppelt nichtnegative Relaxierung ermittelt wurden, unterscheiden sich von den in Tabelle 4.1 gelisteten Werten in der Spalte `SDP OPT`. Dies zeigt, dass einige Nichtnegativitäts-Restriktionen in den Optimallösungen aktiv sind.

Im direkten Vergleich ist das Paket SDPNAL wesentlich schneller als HYBRID, kann aber nicht dessen hohe Genauigkeit erreichen.

$\mathtt{n}^2$	$error^S_{PD}$	`CPU-S`	$error^N_{PD}$	`CPU-N`	$error^H_{PD}$	`CPU-H`
2500	4.8184e-7	58.3	4.3608e-10	23.7	3.7374e-15	187.7
10000	7.9456e-7	2439.9	4.1345e-10	24.3	5.2439e-15	459.8
40000	–	–	5.5432e-10	34.5	7.9604e-15	1968.9
160000	–	–	1.8490e-9	53.5	1.6813e-14	6723.3
490000	–	–	2.7595e-9	187.6	1.6630e-14	24116.8
1000000	–	–	8.1630e-11	496.0	2.3203e-14	67951.8

Tabelle 4.2: SeDuMi vs SDPNAL vs Hybrid

Es gibt natürlich andere äquivalente Relaxierungen des Problems der maximalen stabilen Menge bei Verwendung des doppelt nichtnegativen Kegels. Die Nebenbedingungen

$$(ST) \qquad X \geq 0, \quad X_{i,j} = 0 \ \text{ für alle } i,j \text{ mit } (A_G)_{i,j} > 0$$

könnten beispielsweise in der kürzeren Form

$$(KU) \qquad X \geq 0, \quad A_G \bullet X = 0$$

formuliert werden. Die $\mathcal{S}^n_+ \times \mathcal{N}^n$-Reformulierung beider Versionen verletzt die Slater-Bedingung. Während das Paket SDPNAL mit der kürzeren Form größere Probleme hat, sind die Unterschiede bei der reinen APD-Methode und bei HYBRID wesentlich kleiner. Dies verdeutlicht Tabelle 4.3. Es wurden die in Tabelle 4.1 gelisteten Probleme der Dimensionen 50, 100, 200 und 400 sowohl mit der kurzen Beschreibung (KU) als auch mit der normalen Beschreibung (ST) (vgl. Tabelle 4.2) mit dem reinen APD-Verfahren ($error^A_{PD}$, `CPU-A`), SDPNAL ($error^N_{PD}$, `CPU-N`) und HYBRID ($error^H_{PD}$, `CPU-H`) gelöst. Beim APD-Verfahren wurden jedes Mal 5000 Iterationen berechnet.

Dim	$error_{PD}^{A}$	CPU-A	$error_{PD}^{N}$	CPU-N	$error_{PD}^{H}$	CPU-H
50-KU	4.6887e-5	485.1	4.0633e-10	11.2	5.1911e-14	214.5
50-ST	1.2637e-5	507.5	4.3608e-10	23.7	3.7374e-15	187.7
100-KU	1.6288e-4	1442.1	7.6266e-5	69.9	8.5485e-15	902.1
100-ST	8.5381e-6	1472.2	4.1345e-10	24.3	5.2439e-15	459.8
200-KU	1.7500e-4	5318.1	2.6167e-4	23.7	1.0980e-12	4280.6
200-ST	5.0998e-6	5405.1	5.5432e-10	34.5	7.9604e-15	1968.9
400-KU	1.3487e-4	19716.7	5.6376e-5	1010.2	9.3834e-12	21238.9
400-ST	3.3054e-6	18946.2	1.8490e-9	53.5	1.6813e-14	6723.3

Tabelle 4.3: KURZ vs STANDARD

Zufällige Probleme

Es wurde ein Generator für zufällige Probleme erstellt, welche die Bedingungen (4.2), (4.3) und auch $T_1(X^{opt}) \cap T_2(X^{opt}) = \{0\}$ erfüllen sollen. In Tabelle 4.4 listen wir alle relevanten Daten einiger mit dem Generator erzeugter Probleme auf. Wir vergleichen wieder die drei Löser SeDuMi, SDPNAL und HYBRID. Die Testergebnisse finden sich in Tabelle 4.5. Alle Pakete lieferten für jedes

n^2	m	nnz(A) (%)	DNN OPT
2500	850	82548 (3.8846)	154.138
10000	3367	663667 (1.9711)	2074.593
40000	13400	5319852 (0.9925)	8130.074
160000	26733	1067834 (0.0250)	8935.578

Tabelle 4.4: Zufällige Testprobleme - Problemdaten

n^2	$error_{PD}^{S}$	CPU-S	$error_{PD}^{N}$	CPU-N	$error_{PD}^{H}$	CPU-H
2500	3.6113e-8	63.9	5.1087e-10	3.5	3.2709e-15	331.3
10000	2.4107e-8	4277.9	1.0862e-9	44.3	5.8780e-15	5090.0
40000	–	–	1.4183e-12	406.4	1.1997e-14	25377.3
160000	–	–	5.9576e-12	911.6	1.4347e-14	181049.4

Tabelle 4.5: Zufällige Testprobleme - Ergebnisse

Testbeispiel dieselben Lösungen (bis auf den jeweiligen Fehler), welche jedes Mal strikt komplementär waren. Dies spricht dafür, dass die zu Beginn des Abschnitts erwähnten Voraussetzungen in allen Beispielen erfüllt waren.

Wird die Anzahl der Nebenbedingungen `m` im Generator zu klein gewählt (z.B. $\texttt{m} \leq 15$ für $\texttt{n} = 100$), dann sind die Lösungen nicht mehr strikt komplementär und auch die Zerlegung der dualen Komponente S^{opt} ist nicht eindeutig. HYBRID kann in solchen Fällen unter erhöhtem Zeitaufwand noch immer eine sehr genaue Lösung bestimmen. SDPNAL kommt mit dieser Verletzung der Voraussetzungen jedoch weniger gut zurecht. Bei einem Beispiel mit `n`=100 und `m`=10 hat HYBRID nach etwa 12000 Sekunden eine Lösung mit einem Fehler von 1e-15 berechnet. SDPNAL ist nicht über eine Genauigkeit von 1e-6 hinausgekommen.

Box-QP

Abschließend vergleichen wir SeDuMi, SDPNAL und HYBRID für eine Klasse von speziellen Programmen, welche systematisch die Slater-Bedingung verletzen. Es sind Relaxierungen von BoxQPs – siehe [2]. Durch Betrachtung von 3×3-Untermatrizen jedes zulässigen Punktes der Probleme lässt sich zeigen, dass keine strikt zulässigen Punkte existieren. Damit ist nicht gewährleistet, dass eine Lösung des dualen Programmes überhaupt existiert. Tatsächlich haben alle drei Pakete Schwierigkeiten bei der Lösung dieser Probleme, wie man in Tabelle 4.6 sehen kann. Obwohl die Laufzeit von HYBRID für diese Dimensionen recht hoch

name	$error^S_{PD}$	CPU-S	$error^N_{PD}$	CPU-N	$error^H_{PD}$	CPU-H
spar020-100-1	5.3428e-7	46.2	2.3013e-4	37.0	4.4378e-8	26334.4
spar020-100-2	5.6728e-7	40.5	4.5662e-4	29.9	3.1388e-6	26749.3
spar020-100-3	7.6865e-8	16.9	6.0299e-11	7.5	2.8243e-14	2403.9
spar030-060-1	8.5465e-7	244.4	2.1210e-4	71.4	8.6802e-11	25997.8
spar030-060-2	3.0945e-8	80.8	1.4903e-4	61.7	2.2460e-14	25971.3
spar030-060-3	5.7379e-7	250.8	5.3032e-4	71.2	1.6222e-6	70039.4
spar030-070-1	1.6875e-6	263.4	1.6644e-4	48.5	1.0312e-10	37723.0
spar030-070-2	1.7294e-8	188.7	3.5108e-4	75.3	1.4566e-13	14886.7
spar030-070-3	4.3191e-7	292.8	3.6950e-4	55.8	1.9165e-5	66258.7
spar030-080-1	1.1899e-6	229.7	1.7014e-4	95.8	1.1907e-10	28174.7
spar030-080-2	7.0767e-8	137.4	2.2414e-10	42.2	3.9376e-14	19633.6
spar030-080-3	1.9937e-8	168.5	8.0976e-5	84.3	9.3246e-14	22405.3

Tabelle 4.6: BoxQP Beispiele

ist, konnte es viele Probleme sehr gut lösen – im Gegensatz zu SeDuMi und vor allem SDPNAL. Dies zeigt einerseits eine gewisse Robustheit des HYBRID-Ansatzes und unterstreicht andererseits die Wichtigkeit der Regularität, welche in Abschnitt 4.3 beschrieben wurden.

Fazit

Um die APD-Methode auf den doppelt nichtnegativen Kegel zu erweitern, wurden neue Regularitätsbedingungen eingeführt. Diese stimmen für lineare und semidefinite Programme überein, können sich für beliebige Kegel $\mathcal{K}$ aber sehr wohl unterscheiden. Ist jedoch $\mathcal{K} = DNN^n$ und ist die Optimallösung eindeutig und weiterhin Bedingung (4.3) und $T_1(X^{opt}) \cap T_2(X^{opt})$ erfüllt, dann ist die Verwendung von APD sinnvoll. Um Lösungen höherer Genauigkeit zu erhalten, kann die erweiterte HYBRID-Methode verwendet werden.

4.5 Verallgemeinerte semidefinite Programme

In diesem Abschnitt wollen wir noch einmal die bereits in Abschnitt 2.4 beschriebene Verallgemeinerung von semidefiniten Programmen betrachten:

Sei für $p \in \mathbb{N}$ und $\bar{n} = (n_1, \ldots, n_p) \in \mathbb{N}^p$ der euklidische Raum $E_{\bar{n}} = \mathcal{S}^{n_1} \times \cdots \times \mathcal{S}^{n_p}$ mit dem Skalaprodukt $\langle ., . \rangle_{E_{\bar{n}}}$ und der selbstduale Kegel $\mathcal{K} = \mathcal{S}_+^{n_1} \times \cdots \times \mathcal{S}_+^{n_p}$ gegeben. Das primal-duale Programmpaar ist von der Form

$$(\bar{P}) \quad \min\{\sum_{i=1}^{p} C_i \bullet X_i \mid \mathcal{A}(X_1, \ldots, X_p) = \bar{b},\ X_i \succeq 0 \text{ für } 1 \leq i \leq p\},$$

wobei $\mathcal{A} : E_{\bar{n}} \to I\!R^m$ ein linearer Operator ist, der sich folgendermaßen beschreiben lässt: Es existieren Elemente $A^{(i,j)} \in \mathcal{S}^{n_j}$ für $i = 1, \ldots, m$ und $j = 1, \ldots, p$, so dass

$$\mathcal{A}(X_1, \ldots, X_p) = \begin{pmatrix} \sum_{j=1}^{p} A^{(1,j)} \bullet X_j \\ \vdots \\ \sum_{j=1}^{p} A^{(m,j)} \bullet X_j \end{pmatrix}.$$

Somit hat das duale Problem die Form

$$(\bar{D}) \quad \min\{\bar{b}^T y \mid C_j - \sum_{i=1}^{m} y_i A^{(i,j)} \succeq 0 \text{ für } j = 1, \ldots, p\}.$$

Für $p = 1$ erhalten wir die in Kapitel 2 beschriebenen semidefiniten Programme. Gilt $p = n$ für ein $n \in \mathbb{N}$ und $n_i = 1$ für $1 \leq i \leq n$, dann liegt ein lineares Programm vor. Wählen wir für ein $k \in \mathbb{N}$ beispielsweise $p = k^2 + 1$ und $n_1 = k$ sowie $n_i = 1$ für $2 \leq i \leq k^2 + 1$, dann erhalten wir bei entsprechender Wahl von $\mathcal{A}$ die in Abschnitt 4.1 betrachteten DNN-Reformulierungen $(\tilde{P})$ und $(\tilde{D})$. Wir können allgemein jedes konische Programm beschreiben, dessen Kegel eine Kombination von semidefiniten (und linearen) Blöcken ist. Die in Abschnitt 1.3 definierte Funktion $\tilde{\phi}$ hat die Form:

$$\tilde{\phi}((X_1, \ldots, X_p), (S_1, \ldots, S_p)) = \frac{1}{2} \sum_{j=1}^{p} \|X_j - \Pi_{\mathcal{S}_+^{n_j}}(X_j)\|_F^2 + \|S_j - \Pi_{\mathcal{S}_+^{n_j}}(S_j)\|_F^2.$$

Sie besitzt alle relevanten Eigenschaften, die auch die Version für den Fall $p = 1$ besitzt (siehe Kapitel 2). Die Funktion

$$\tilde{f}_{\bar{n}}((X_1, \dots X_p), (S_1, \dots, S_p)) = \sum_{j=1}^{p} \|X_j S_j\|_F^2$$

erfüllt nach Satz 11 für eine Optimallösung

$$Z_{\bar{n}}^{opt} = ((X_1^{opt}, \dots, X_p^{opt}), (S_1^{opt}, \dots, S_p^{opt}))$$

die Eigenschaft

$$\nabla^2 \tilde{f}_{\bar{n}}(Z_{\bar{n}}^{opt}) \succ 0,$$

sofern die Optimallösung eindeutig ist und $X_j^{opt} + S_j^{opt} \succ 0$ für $1 \leq j \leq p$ gilt. (Dies entspricht Voraussetzung 3.) Es gilt damit auch in diesem allgemeinen Fall

$$\partial^2(\tilde{\phi} + \tilde{f}_{\bar{n}})(Z_{\bar{n}}^{opt}) \succ 0, \tag{4.8}$$

wobei die Herleitung analog zum Fall $p = 1$ verläuft. Unter Voraussetzung 3 konvergiert das verallgemeinerte Newton-Verfahren nach Satz 4 somit lokal quadratisch. Eine Verallgemeinerung der in Abschnitt 2.5 vorgestellten Variante des APD-Verfahrens zur Bestimmung der Optimallösung ist somit sinnvoll und analog zum Fall $p = 1$ implementierbar.

Abschließend gehen wir noch darauf ein, wie das in Abschnitt 3.2 vorgestellte AHO-QMR-Verfahren auf $\mathcal{K} = \mathcal{S}_+^{n_1} \times \dots \times \mathcal{S}_+^{n_p}$ verallgemeinert werden kann: Mit Satz 1 kann gefolgert werden, dass jede Lösung $((X_1, \dots, X_p), y, (S_1, \dots, S_p))$ des folgenden Systems eine Optimallösung von $(\bar{P})$ und $(\bar{D})$ ist.

$$\begin{array}{rcl}
\sum_{j=1}^{p} A^{(1,j)} \bullet X_j & = & \bar{b}_1 \\
\vdots & & \vdots \\
\sum_{j=1}^{p} A^{(m,j)} \bullet X_j & = & \bar{b}_m \\
\\
\sum_{i=1}^{m} y_i A^{(i,1)} + S_i & = & C_1 \\
\vdots & & \vdots \\
\sum_{i=1}^{m} y_i A^{(i,p)} + S_i & = & C_p \\
\\
X_1 S_1 & = & 0 \\
\vdots & & \vdots \\
X_p S_p & = & 0 \\
\\
X_1, \dots, X_p & \succeq & 0 \\
S_1, \dots, S_p & \succeq & 0
\end{array}$$

Wie im Fall $p = 1$ kann die (verallgemeinerte) AHO-Richtung durch Anpassung der Komplementaritätsgleichungen $X_i S_i = 0$ zu $X_i S_i + S_i X_i = 0$ und eine

anschließende Linearisierung des Systems bestimmt werden. Somit ist auch das AHO-QMR-Verfahren anwendbar, sofern eine geeigente Strategie zur Zerlegung der Variablen in einen zu elimierbaren und einen nicht zu eliminierbaren Anteil gewählt wird (siehe Abschnitt 3.2). Es ist weiterhin mit minimalem Mehraufwand möglich, die rechte Seite des zu lösenden AHO-Systems durch einen Parameter $\mu > 0$ zu stören um den aus den Innere-Punkte-Methoden bekannten „zentralen Pfad" anzunähren, anstatt direkt die Optimallösung zu approximieren.

Literaturverzeichnis

[1] Farid Alizadeh, Jean-Pierre A. Heaberly und Michael L. Overton, Primal-Dual Interior-Point Methods for Semidefinite Programming: Convergence Rates, Stability and Numerical Results, SIAM Journal on Optimization, 1998, vol. 8, (3), 746–768.

[2] Samuel Burer, On the copositive representation of binary and continuous nonconvex quadratic programs, Mathematical Programming, 2009, vol. 120, (2), 479–495.

[3] Thomas Davi und Florian Jarre, High accuracy solutions of large scale semidefinite programs, Optmization Methods and Software, 2012, vol. 27.

[4] Thomas Davi und Florian Jarre, On the stable solution of large scale problems over the doubly nonnegative cone, eingereicht bei Mathematical Programming, 2012

[5] James W. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, 1997.

[6] DIMACS Implementation Challenges,
ftp://dimacs.rutgers.edu/pub/challenge/graph

[7] Roland Freund und Noel Nachtigal, QMR: a Quasi-Minimal Residual Method for Non-Hermitian Linear Systems, Numerische Mathematik, 1991, vol. 60, (1), 315–339.

[8] Chantal Hergenröder, Die erweiterte Primal-Dual-Funktion für semidefinite Programme, Diplomarbeit, 2008.

[9] Florian Jarre und Franz Rendl, An Augmented Primal-Dual Method for Linear Conic Programs, SIAM Journal on Optimization, 2008, vol. 19, (2), 808–823.

[10] Florian Jarre und Josef Stoer, Optimierung, Springer-Verlag, Berlin Heidelberg, 2004

[11] Laslo Lovasz, On the Shannon Capacity of a Graph, IEEE Transactions on Information Theory, 1979, vol. 25, 1–7.

[12] Jerome Malick, Janez Povh, Franz Rendl und Angelika Wiegele, Regularization methods for semidefinite programming, SIAM Journal on Optimization, 2009, vol. 20, (1), 336–356.

[13] Yasuaki Matsukawa und Akiko Yoshise, On optimization over the doubly nonnegative cone, CACSD, 2010, 13–18.

[14] Katta G. Murty und Santosh N. Kabadi, Some NP-complete problems in quadratic and linear programming, Mathematical Programming, 1987, vol. 39, 117–129.

[15] Arkadii Nemirovskii und Yurii Nesterov, Interior-Point Polynomial Algorithms in Convex Programming, SIAM, Philadelphia, 1994.

[16] Jorge Nocedal und Stephen J. Wright, Numerical Optimization, Springer-Verlag, New York, 1999.

[17] Chris C. Paige und Michael A. Saunders, Solution of sparse indefinite systems of linear equations, SINUM, 1975, vol. 12, 617–629.

[18] Imre Polik, SeDuMi, 2011, Download from http://sedumi.ie.lehigh.edu/

[19] Florian A. Potra und Rongqin Sheng, A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming, SIAM Journal on Optimization, 1998, vol. 8, (4), 1007–1028.

[20] Liqun Qi und Jie Sun, A nonsmooth version of Newton's method, Mathematical Programming, 1993, vol. 58, 353–367.

[21] Hans Rademacher, Über partielle und totale Differenzierbarkeit von Funktionen mehrerer Variabeln und über die Transformation der Doppelintegrale, Math. Ann., 1919, vol. 79, 340-359.

[22] Franz Rendl, Random Matrix Generator, rand_sdps.m, Download from http://www.math.uni-klu.ac.at/or/Software/software.html

[23] Katrin Schmallowsky, On the Regularity of Second Order Cone Programs and an Application to Solving Large Scale Problems, Mathematical Methods of Operations Research, 2009, vol. 68, (3), 551-564.

[24] Jos Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization Methods and Software, 1999, vol. 11-12, 625–653.

[25] Defeng Sun und Jie Sun, Semismooth Matrix-Valued Functions, Mathematics of Operations Research, 2002, vol. 27, (1), 150–169,

[26] Xin-Yuan Zhao, Defeng Sun und Kim-Chuan Toh, A Newton-CG augmented Lagrangian method for semidefinite programming, Technical report, National University of Singapore, 2008.

Hiermit versichere ich, die vorliegende Dissertation selbständig und ohne unerlaubte Hilfsmittel erstellt zu haben. Die Dissertation wurde bisher weder in dieser noch in ähnlicher Form bei einer anderen Institution eingereicht. Ich habe zuvor noch keinen Promotionsversuch unternommen.

Thomas Davi Düsseldorf, den 2.Mai.2012