

Universität
Rostock



Traditio et Innovatio

Ein Simplexalgorithmus für rationale CP -Faktorisierung

Masterarbeit

Mathematisch-Naturwissenschaftliche Fakultät

Institut für Mathematik

Name:	Valentin Florian Dannenberg
Matrikelnummer:	215204272
Betreuer und Gutachter:	Prof. Dr. rer. nat. habil. Achill Schürmann
Gutachter:	Dr. rer. nat. habil. Frieder Ladisch
Abgabedatum:	11.11.2020

Inhaltsverzeichnis

Symbolverzeichnis	III
1 Einleitung	1
1.1 Einleitung	1
1.2 Übersicht	2
2 Die Kegel COP^d und CP^d - Grundlagen, Anwendungen und offene Probleme	3
2.1 Grundlegende Definitionen	3
2.2 Kegel, Dualität und Eigenschaften von CP^d und COP^d	4
2.2.1 Kegel und einige Resultate über CP^d und COP^d	4
2.2.2 Dualität	6
2.2.3 COP^d und CP^d für kleine Dimensionen	8
2.2.4 Zusätzliche Definitionen	9
2.3 Anwendungen	10
2.3.1 Konische Optimierung	10
2.3.2 Umformulierungen bekannter schwerer Optimierungsprobleme	12
2.3.3 Polyedrische Approximation von CP^d und COP^d	16
2.4 Die Extremalstrahlen von COP^d	19
2.4.1 Minimale Nullen einer kopositiven Matrix	19
2.4.2 Bestimmung der Extremalstrahlen von COP^d	21
2.4.3 Maximale Seiten von CP^d	22
2.5 COP -perfekte Matrizen	25
3 Ein Simplexalgorithmus für rationale CP-Faktorisierung	29
3.1 Termination des Algorithmus	31
3.2 Analyse von Algorithmus 1	34
3.2.1 Bestimmung eines geeigneten Extremalstrahles von $(\mathcal{V}(P))^*$	35
3.2.2 Bestimmung einer benachbarten COP -perfekten Matrix	45
3.2.3 Kopositivitätstests	49
4 Ein möglicher Algorithmus zum Lösen des CP-Membership-Problems	55
4.1 Das CP -Membership-Problem	55
4.2 Modifikation von Algorithmus 1	56
4.3 Termination in Dimension 2	60
4.4 3×3 und 4×4 COP -perfekte Matrizen	62

5	Das Kpositive Minimum	65
5.1	Klassische Berechnung des kpositiven Minimums	65
5.1.1	Simplexpartitionierungen	66
5.1.2	Hermitsche Normalform	70
5.1.3	Berechnung der Koeffizienten α_i	72
5.2	Ein experimenteller Algorithmus	75
5.2.1	Erste Analyse des quadratischen Problems	75
6	Fazit und Ausblick	79

Symbolverzeichnis

\mathbb{R}^d - Reeller Vektorraum der Dimension d

$\mathbb{R}_{\geq 0}^d$ - Nichtnegativer Orthant des \mathbb{R}^d

\mathcal{S}^d - Vektorraum der reellen symmetrischen $d \times d$ -Matrizen

\mathcal{CP}^d - Kegel der vollständig positiven $d \times d$ Matrizen

\mathcal{COP}^d - Kegel der kopositiven $d \times d$ Matrizen

$\mathcal{S}_{\geq 0}^d$ - Kegel der positiv semidefiniten $d \times d$ Matrizen

$\mathcal{S}_{> 0}^d$ - Kegel der positiv definiten $d \times d$ Matrizen

$\text{conv}M$ - Konvexe Hülle einer Menge M

$\text{cone}M$ - Konische Hülle einer Menge M

$\text{int}M$ - Innere einer Menge M

$\text{bd}M$ - Rand einer Menge M

$\text{cl}M$ - Abschluss einer Menge M

$\text{rk}A$ - Rang einer Matrix A

I - Identitätsmatrix

E - Einsmatrix

e_i - i -ter Einheitsvektor

e - Einsvektor

K^* - Zu K dualer Kegel

$\mathcal{V}(P)$ - Voronoi-Kegel der Matrix P

1 Einleitung

1.1 Einleitung

Eine reelle symmetrische Matrix Q heißt *vollständig positiv*, wenn sie sich als Summe nichtnegativer Rang-1-Matrizen xx^T schreiben lässt. Geometrisch lassen sich solche Matrizen folgendermaßen motivieren: Angenommen wir hätten n reelle Vektoren v_1, \dots, v_n der Länge d .

Gibt es eine lineare Transformation, die diese Vektoren in den nichtnegativen Orthanten $\mathbb{R}_{\geq 0}^d$ einbettet?

Die Antwort auf diese Frage lautet Ja, wenn die Gram-Matrix Q mit $Q_{ij} = v_i^T v_j$ vollständig positiv ist [14].

Stark mit den vollständig positiven Matrizen zusammenhängend sind die von Motzkin 1952 [91] „getauften“ *konditionellen positiv semidefiniten* Matrizen, heutzutage auch *kopositiv* genannt. Dabei heißt eine reelle symmetrische Matrix Q kopositiv, falls

$$v^T Q v \geq 0 \text{ für alle komponentenweise nichtnegativen } v \text{ ist.}$$

Die Mengen dieser beiden Matrizenarten sind für geometrische Untersuchungen sehr interessant, da beide Mengen sogenannte *Kegel*, mit vielen interessanten Eigenschaften, bilden. Auch tauchen diese Matrizen heutzutage in einer Vielzahl von Anwendungen in den verschiedensten Bereichen auf, so lassen sich beispielsweise Evolutionsmodelle durch vollständig positive Matrizen testen [76]. Doch die wohl bekannteste Anwendung der vollständig positiven und kopositiven Matrizen ist, dass man mittels diesen sehr schwierige quadratische Optimierungsprobleme als sogenannte „konische Optimierungsprobleme“ darstellen kann [20]. Dies zeigte, dass es sehr schwere konvexe Optimierungsprobleme gibt, wovon man bis dahin nicht ausgegangen ist.

Ein für die Anwendung sehr wichtiges, noch offenes, Problem ist, entscheiden zu können, ob eine gegebene Matrix vollständig positiv ist oder nicht. Kürzlich konnte von Dickinson und Gijben gezeigt werden, dass die algorithmische Lösung dieses Problems \mathcal{NP} -schwer ist [43]. Über die Jahre wurden die verschiedensten Algorithmen entwickelt, um dieses Problem anzugehen, dennoch gibt es bisher keinen Algorithmus, dem dies gelungen ist.

In dieser Arbeit wird der von Dutour Sikirić, Schürmann und Vallentin in [47] entwickelte Algorithmus, welcher versucht das obige Problem zu lösen, untersucht. Dabei stellen wir diesen Algorithmus mit den notwendigen Grundlagen aus dem Bereich der

konvexen und diskreten Geometrie detailliert und mit allen, für eine Implementierung notwendigen, Komponenten vor. Weiter zeigen wir, dass er das Problem für kleine Dimensionen tatsächlich löst und gehen auf offene Fragen und Probleme rund um vollständig positive Matrizen, kopositive Matrizen und den Algorithmus ein.

1.2 Übersicht

Zunächst geben wir in Kapitel 2 in den Sektionen 2.1 und 2.2.1 die wichtigsten Definitionen für diese Arbeit und einige allgemeine Eigenschaften der kopositiven und vollständig positiven Matrizen an. In Sektion 2.3 folgen nach einer kurzen Einführung in die konische Optimierung, Anwendungen der vollständig positiven Matrizen in diesem Bereich und welche Vor- und Nachteile diese mit sich bringen. Ein noch offenes Problem der kopositiven Matrizen betrachten wir in Sektion 2.4 und geben momentane Entwicklungen und Resultate diesbezüglich an. Den Abschluss dieses Kapitels bildet Sektion 2.5, in der wir die spezielle Klasse der \mathcal{COP} -perfekten Matrizen einführen werden.

In Kapitel 3 widmen wir uns dem von Dutour Sikirić, Schürmann und Vallentin in [47] entwickeltem Simplexalgorithmus für \mathcal{CP} -Faktorisierungen. Dabei gehen wir zunächst in Sektion 3.1 auf seine Funktionsweise ein und geben den Beweis für die endliche Termination aus [47] an. Die einzelnen Schritte des Algorithmus werden wir in Sektion 3.2 analysieren, sowie alle benötigten Grundlagen für eine Implementation kennenlernen. Dabei befassen wir uns in Untersektion 3.2.1 mit der algorithmischen Bestimmungen der Extremalstrahlen eines Kegels, in Untersektion 3.2.2 mit der Bestimmung benachbarter \mathcal{COP} -perfekten Matrizen und in Sektion 3.2.3 mit algorithmischen Kopositivitätstests.

Der in [47] entwickelte Algorithmus zum Lösen des \mathcal{CP} -Membership-Problems steht in Kapitel 4 im Fokus. Hier betrachten wir erst in Sektion 4.1 das genannte Problem sowie momentane Entwicklungen und gehen dann in Sektion 4.2 auf den Algorithmus aus [47] ein. Dabei besprechen wir die Korrektheit des Algorithmus und geben in Theorem 4.1 eigene Verbesserungen an. In Sektion 4.3 beweisen wir [47, Vermutung 3.2] für den Fall $d = 2$ und gehen in Sektion 4.4 auf die Fälle $d \geq 3$ ein.

Die Berechnung des kopositiven Minimums einer Matrix besprechen wir in Kapitel 5. Dazu geben wir in Sektion 5.1 den „klassischen“, in [47], entwickelten Ansatz an. Danach präsentieren wir in Sektion 5.2 einen neuen Ansatz von Schürmann vor, welcher in Teilen noch genauer untersucht werden muss. Dafür geben wir in Untersektion 5.2.1 erste Ideen an.

2 Die Kegel \mathcal{COP}^d und \mathcal{CP}^d - Grundlagen, Anwendungen und offene Probleme

2.1 Grundlegende Definitionen

Zunächst definieren wir über $v \geq w$ genau dann, wenn $v_i \geq w_i$ ist, eine Halbordnung auf dem reellen d -dimensionalen Vektorraum \mathbb{R}^d . Mit dieser Halbordnung definieren wir dann den *nichtnegativen Orthanten* von \mathbb{R}^d als

$$\mathbb{R}_{\geq 0}^d = \{v \in \mathbb{R}^d \mid v \geq 0\}.$$

Den *Vektorraum der reellen symmetrischen $d \times d$ -Matrizen* bezeichnen wir mit \mathcal{S}^d . Dieser wird zusammen mit dem Skalarprodukt $\langle Q, B \rangle = \text{Spur}(QB)$ ein euklidischer Vektorraum. Über das Skalarprodukt können wir dann auch eine Norm $|Q| = \sqrt{\langle Q, Q \rangle}$ und Topologie, welche durch die Mengen

$$B(Q, \varepsilon) = \{B \in \mathcal{S}^d \mid |Q - B| < \varepsilon\}$$

erzeugt wird, auf \mathcal{S}^d definieren. Dies ermöglicht es uns, über das Innere $\text{int}S$ und den Rand $\text{bd}S$ einer Menge S zu sprechen.

Die *Menge der vollständig positiven $d \times d$ -Matrizen* definieren wir als

$$\mathcal{CP}^d = \left\{ \sum_{i=1}^n v_i v_i^T \mid v_i \in \mathbb{R}_{\geq 0}^d \right\}.$$

Mit der Kurzbezeichnung $Q[v] = \langle Q, vv^T \rangle = v^T Q v$ bezeichnen wir dann die *Menge der kopositiven $d \times d$ -Matrizen* als

$$\mathcal{COP}^d = \{Q \in \mathcal{S}^d \mid Q[v] \geq 0 \text{ für alle } v \in \mathbb{R}_{\geq 0}^d\}.$$

Für ein Element a eines reellen euklidischen Vektorraums V mit Skalarprodukt $\langle \cdot, \cdot \rangle$ und einer reellen Zahl β nennen wir die Menge

$$H(a, \beta) = \{x \in V \mid \langle a, x \rangle = \beta\}$$

Hyperebene (mit Normalenvektor a). Ersetzt man in der obigen Menge das Gleichheitszeichen durch \geq oder \leq , so nennen wir die entstehende Menge auch *Halbraum*.

Im Folgenden verwenden wir weiterhin folgende Notation für einige spezielle Matrizen und Vektoren. Mit e_i bezeichnen wir den i -ten Einheitsvektor passender Dimension, mit e den Vektor $(1, \dots, 1)$. Die Einheitsmatrix passender Dimension wird mit I notiert sowie die Einsmatrix mit $E = ee^T$.

2.2 Kegel, Dualität und Eigenschaften von \mathcal{CP}^d und \mathcal{COP}^d

In dieser Sektion geben wir eine Einführung und Übersicht über die wichtigsten Begriffe und Konzepte, die in dieser Arbeit vorkommen. Da wir diese Themen hier natürlich nicht zur Gänze darstellen können, verweisen wir auf [8].

2.2.1 Kegel und einige Resultate über \mathcal{CP}^d und \mathcal{COP}^d

Eine Menge K eines reellen Vektorraums V wird *Kegel* genannt, falls für alle $v, w \in K$ und alle $\lambda, \mu \in \mathbb{R}_{\geq 0}$ auch $\lambda v + \mu w \in K$ ist. Die *konische Hülle einer Menge* S , die Menge aller nichtnegativen Linearkombinationen von Elementen aus S , bezeichnen wir mit $\text{cone}M$.

Die beiden wichtigsten Objekte dieser Arbeit \mathcal{CP}^d und \mathcal{COP}^d sind, wie man leicht sieht, Kegel, mit einigen weiteren Eigenschaften, die wir im Folgenden darlegen werden. Weitere Kegel, die für diese Arbeit von Bedeutung sind, sind

$$\mathcal{N}^d = \{Q \in \mathcal{S}^d \mid q_{ij} \geq 0, \text{ für alle } i, j = 1, \dots, d\}$$

der Kegel der nichtnegativen Matrizen und

$$\mathcal{S}_{\geq 0}^d = \{Q \in \mathcal{S}^d \mid Q[x] \geq 0 \text{ für alle } x \in \mathbb{R}^d\}$$

der Kegel der positiv semidefiniten Matrizen. Das Innere von $\mathcal{S}_{\geq 0}^d$, den Kegel der positiv definiten Matrizen, bezeichnen wir mit $\mathcal{S}_{> 0}^d$. Offenbar erfüllt jede positiv semidefinite und jede nichtnegative Matrix Q , dass $Q[x] \geq 0$ für alle $\mathbb{R}_{\geq 0}^d$ ist, sodass $\mathcal{S}_{\geq 0}^d, \mathcal{N}^d \subseteq \mathcal{COP}^d$ ist. Da die kopositiven Matrizen einen Kegel bilden, erhalten wir daher $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d \subseteq \mathcal{COP}^d$. Weiter sehen wir auch, dass jede vollständig positive Matrix nur nicht negative Einträge enthält und positiv semidefinit ist, daher haben wir $\mathcal{CP}^d \subseteq \mathcal{S}_{\geq 0}^d \cap \mathcal{N}^d$. In Sektion 2.2.3 betrachten wir diese Inklusionen für kleine Dimensionen und zeigen, dass sie ab einer gewissen Dimension echt sind. Die kopositiven Matrizen bilden also im Bezug auf die positiv semidefiniten und nichtnegativen Matrizen einen „sehr großen“ Kegel, die vollständig positiven einen eher „kleinen“ Kegel.

Alle oben genannten Kegel sind abgeschlossen, volldimensional und punktiert. Dabei heißt ein Kegel K *punktiert*, wenn er keine Gerade enthält, also $K \cap (-K) = \{0\}$. Man beachte, dass es ausreicht diese drei Eigenschaften für \mathcal{CP}^d oder \mathcal{COP}^d nachzuweisen, wie wir in der nächsten Sektion sehen werden. Kurz werden solche Kegel auch *ordentlich* genannt.

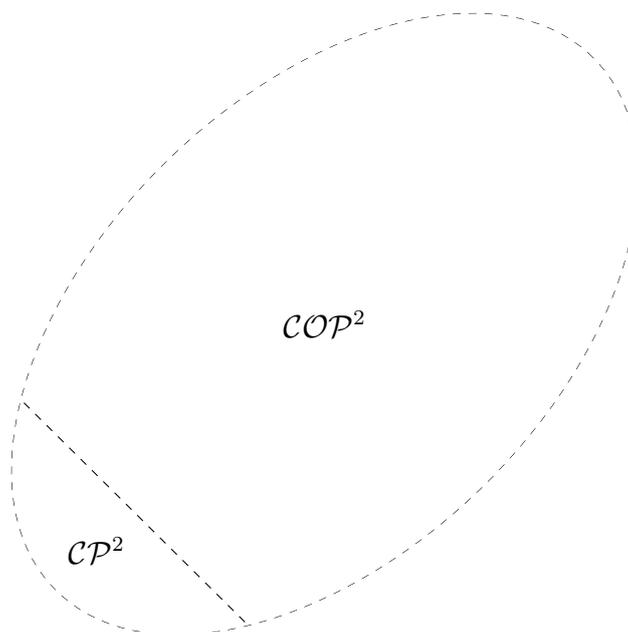


Abbildung 2.1: Schnitt einer Hyperebene mit \mathcal{COP}^d und \mathcal{CP}^d , siehe [41, Abbildung 8.1]

Für ein $v \in V$ nennen wir die Menge $\text{cone}(v) = \{\lambda v \mid \lambda \geq 0\}$ den *Strahl*, der von v erzeugt wird. Ein Kegel K enthält also alle Strahlen, die von Elementen in K erzeugt werden. Ist $v \neq 0$ ein Element aus K , so dass für alle $u, w \in K$ mit $v = u + w$ gilt, dass $u, w \in \text{cone}(v)$ so nennen wir den Strahl, der von v erzeugt wird, *Extremalstrahl* von K und v *Erzeuger eines Extremalstrahlen* von K [14, Definition 1.24]. Im Folgenden werden wir Erzeuger von Extremalstrahlen auch als Extremalstrahlen bezeichnen.

Da sich jede vollständig positive Matrix als Summe von nichtnegativen Rang-1-Matrizen schreiben lässt, sind die Extremalstrahlen von \mathcal{CP}^d gerade die Matrizen xx^T mit $x \geq 0$ [14, Bemerkung 2.3]. Mit den Extremalstrahlen von \mathcal{COP}^d werden wir uns in Sektion 2.4 beschäftigen, da eine vollständige Liste derer noch ein offenes Problem ist. Einige bekannte Beispiele sind allerdings die Matrizen $E_{ij} = e_i e_j^T + e_j e_i^T$ und xx^T für $x \in \mathbb{R}^d \setminus (\mathbb{R}_{\geq 0}^d \cup (-\mathbb{R}_{\geq 0}^d))$ [14, Proposition 1.25].

Ein Kegel K heißt *polyedrisch*, falls er endlich erzeugt wird, also $K = \text{cone}S$ mit $|S| < \infty$ [14, Definition 1.27] oder äquivalent er der Schnitt von endlich vielen Halbräumen ist, also $K = \bigcap_{i=1}^n H_i^{\geq}(a_i, \beta_i)$ [14, Theorem 1.38]. Ein Beispiel für einen polyedrischen Kegel ist \mathcal{N}^d , der durch die Matrizen E_{ij} erzeugt wird.

In Sektion 3.2.1 benutzen wir, dass ein polyedrischer Kegel K punktiert ist, wenn die Menge $\{a_1, \dots, a_n\}$ volldimensional ist [86, Kapitel 8].

\mathcal{CP}^d und \mathcal{COP}^d sind nicht polyedrisch, da sie unendlich viele Extremalstrahlen besitzen und daher nicht endlich erzeugt werden können. Es gibt aber in beiden Kegeln polyedrische „Abschnitte“ [48], wie wir an unterschiedlichen Stellen in dieser Arbeit sehen werden.

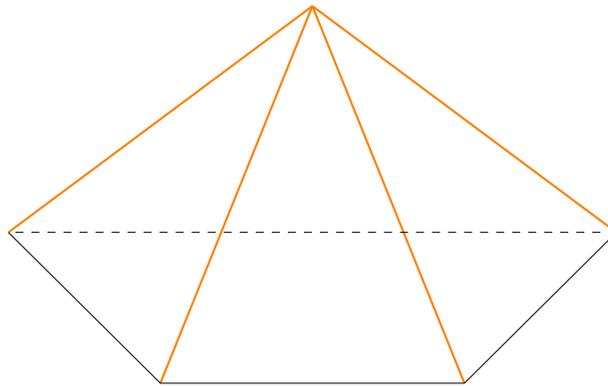


Abbildung 2.2: Polyedrischer Kegel mit 4 Extremlinien (orange)

Als *Symmetriegruppe* $\text{Sym}(K)$ eines Kegels K bezeichnen wir die Menge der Automorphismen unter denen K erhalten bleibt, also

$$\text{Sym}(K) = \{L \in \text{Aut}(V) \mid L(K) = K\}.$$

Da Permutationsmatrizen P und nichtnegative nicht-singuläre Diagonalmatrizen D den nichtnegativen Orthanten $\mathbb{R}_{\geq 0}^d$ erhalten, also $P D x \in \mathbb{R}_{\geq 0}^d$ für alle $x \in \mathbb{R}^d$, bleibt \mathcal{COP}^d unter Abbildungen der Form $P^T D X D P$ erhalten. In [116] wurde gezeigt, dass dies tatsächlich alle Automorphismen von \mathcal{S}^d mit dieser Eigenschaft sind. Wir erhalten also

$$\begin{aligned} \text{Sym}(\mathcal{COP}^d) = \{L \in \text{Aut}(\mathcal{S}^d) \mid L(X) = P^T D X D P \text{ für Permutationmatrix } P \text{ und} \\ D = \text{diag}(a_1, \dots, a_d) \in \mathcal{N}^d, \text{rk} D = d\}. \end{aligned} \quad (2.1)$$

Für \mathcal{CP}^d erhalten wir durch Eigenschaften, die wir in der nächsten Sektion einführen werden und des hier verwendeten Skalarprodukts, dass $\text{Sym}(\mathcal{CP}^d) = \text{Sym}(\mathcal{COP}^d)$.

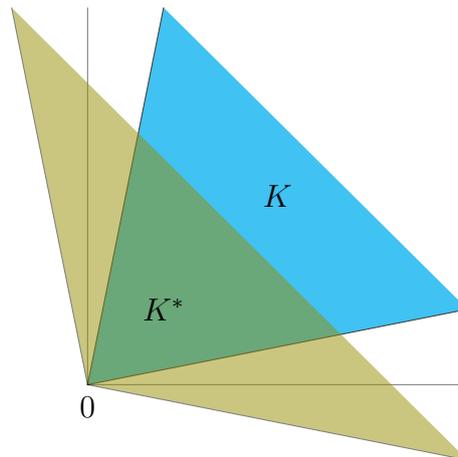
2.2.2 Dualität

Ein weiteres nützliches Hilfsmittel in der Studie von Kegeln ist die Dualität ebensolcher. Sei V ein euklidischer Vektorraum mit Skalarprodukt $\langle \cdot, \cdot \rangle$ und K, C Kegel in V . Dann heißt C *dual* zu K , wenn [8, 5.3]

$$C = \{v \in V \mid \langle v, k \rangle \geq 0 \text{ für alle } k \in K\}.$$

Wir bezeichnen dann C auch mit K^* . Falls $K = K^*$ ist, nennen wir K selbstdual. Beispiele für selbstduale Kegel sind $\mathcal{S}_{\geq 0}^d$ und \mathcal{N}^d . Einige nützliche Eigenschaften von Dualität sind unter anderem

- Für einen Kegel K ist $K^* = (\text{cl} K)^*$. [14, Theorem 1.35]
- Ist K abgeschlossen, so ist $(K^*)^* = K$. [10, Theorem 2.3.1 (iv)]



- K ist genau dann ein ordentlicher Kegel, wenn K^* einer ist. [10, Korollar 2.3.1]
- $\text{int}K = \{k \in K \mid \langle k, x \rangle > 0 \text{ für alle } x \in K^* \setminus \{0\}\}$. [81, Lemma 3.2.4]

Es gilt

$$(\mathcal{CP}^d)^* = \mathcal{COP}^d \text{ und } (\mathcal{COP}^d)^* = \mathcal{CP}^d, \quad (2.2)$$

was sofort aus den Definitionen folgt.

Damit lassen sich nun durch [81, Lemma 3.2.4] Beschreibungen für das Innere und den Rand der beiden Mengen angeben. So ist

$$\text{int } \mathcal{COP}^d = \{Q \in \mathcal{S}^d \mid Q[x] > 0 \text{ für alle } x \in \mathbb{R}_{\geq 0}^d \setminus \{0\}\} \quad (2.3)$$

und

$$\text{bd } \mathcal{COP}^d = \{Q \in \mathcal{S}^d \mid Q[x] = 0 \text{ für ein } x \in \mathbb{R}_{\geq 0}^d \setminus \{0\}\}. \quad (2.4)$$

Die Matrizen im Inneren von \mathcal{COP}^d nennen wir auch *strikt kopositiv*.

Über \mathcal{CP}^d lassen sich ähnliche Aussagen über das Innere und den Rand machen, allerdings sind wirklich anschauliche Beschreibungen des Inneren etwas komplizierter. So ist beispielsweise [49, Theorem 2.3]

$$\text{int } \mathcal{CP}^d = \{Q \in \mathcal{S}^d \mid Q = BB^T, \text{ mit } B = [B_1|B_2], \text{ sodass } B_1 > 0 \text{ und } \text{rk}B_2 = d\}.$$

Eine weitere Darstellung wurde von Dickinson in [41, Theorem 7.4] gefunden. Es ist

$$\text{int } \mathcal{CP}^d = \{Q \in \mathcal{S}^d \mid Q = BB^T, \text{ mit } B = [a|C], \text{ sodass } \text{rk}B = d, a \in \mathbb{R}_{> 0}^d \text{ und } C \geq 0\}.$$

Der Rand von \mathcal{CP}^d ist allerdings noch nicht gut verstanden. Ein Grund dafür wird in Sektion 2.4 präsentiert.

2.2.3 \mathcal{COP}^d und \mathcal{CP}^d für kleine Dimensionen

Wir erinnern, dass für \mathcal{CP}^d und \mathcal{COP}^d die Inklusionen

$$\mathcal{CP}^d \subseteq \mathcal{S}_{\geq 0}^d \cap \mathcal{N}^d \text{ und } \mathcal{S}_{\geq 0}^d + \mathcal{N}^d \subseteq \mathcal{COP}^d \quad (2.5)$$

gelten. Diananda zeigte[39], dass die umgekehrten Inklusionen für $d \leq 4$ gelten. In [14, Sektion 2.3] wird ein geometrischer Beweis hierfür präsentiert, basierend auf der in der Einführung präsentierten geometrischen Interpretation von vollständig positiven Matrizen. Man beachte hierbei, dass wegen (2.2) nur eine der beiden Inklusionen gezeigt werden muss, siehe dazu auch [14, Theorem 1.35].

Für $d \geq 5$ hingegen sind die obigen Inklusionen echt. Dafür betrachten wir für \mathcal{COP}^5 die Hornmatrix [58, Gleichung (16.2.28)]

$$H = \begin{pmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix}.$$

Dass diese Matrix tatsächlich kopositiv ist, wird aus der folgenden Darstellung von $H[x]$ klar. Es ist [58, Gleichung (16.2.29)]

$$\begin{aligned} H[x] &= (x_1 - x_2 + x_3 + x_4 - x_5)^2 + 4(x_2x_4 + x_3(x_5 - x_4)) \\ &= (x_1 - x_2 + x_3 - x_4 + x_5)^2 + 4(x_2x_5 + x_1(x_4 - x_5)). \end{aligned} \quad (2.6)$$

Damit ist H kopositiv, da für $x_4 \geq x_5$ die zweite Gleichung, für $x_5 \geq x_4$ die erste Gleichung nichtnegativ ist. H aber nicht positiv semidefinit und liegt auch nicht in \mathcal{N}^d . Es kann weiter mit (2.6) gezeigt werden, dass H ein Extremalstrahl von \mathcal{COP}^5 ist [58, Seite 357-358] und sich damit nicht als Summe einer positiv semidefiniten und einer nichtnegativen Matrix schreiben lässt. Damit ist also $\mathcal{S}_{\geq 0}^5 + \mathcal{N}^5 \subset \mathcal{COP}^5$. Ist nun R ein Extremalstrahl von \mathcal{COP}^d , so erhalten wir mit

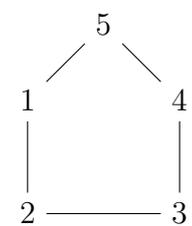
$$\begin{pmatrix} R & 0 \\ 0^T & 0 \end{pmatrix} \quad (2.7)$$

einen Extremalstrahl von \mathcal{COP}^{d+1} [9][41, Theorem 8.20]. Man beachte, dass wir mit (2.1) diese „Null“-Operation an einer beliebigen Stelle durchführen können. Damit lässt sich aus H eine Matrix in $\mathcal{COP}^d \setminus (\mathcal{S}_{\geq 0}^d + \mathcal{N}^d)$ für beliebiges $d \geq 5$ konstruieren. Eine weitere Methode, wie man solche kopositiven Matrizen außerhalb von $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ konstruieren kann, wird in [70] präsentiert.

Für \mathcal{CP}^d geben wir zunächst folgendes graphentheoretisches Argument: Wir definieren den Graphen einer (symmetrischen) Matrix Q als $G(Q) = (V(Q), E(Q))$ mit $V(Q) = \{1, \dots, d\}$ und $E(Q) = \{(i, j) \mid i \neq j, a_{ij} \neq 0\}$ und nennen einen Graphen G vollständig positiv, falls jede Matrix $Q \in \mathcal{S}_{\geq 0}^d \cap \mathcal{N}^d$ mit $G(Q) = G$ vollständig positiv ist. Kogan und Berman zeigten in [77, Theorem 2.1], dass ein Graph genau dann vollständig positiv ist,

wenn er keinen ungeraden Kreis mit Länge > 4 enthält. Offenbar gilt dies für Graphen mit $|V| \leq 4$ immer. Man beachte, dass diese Eigenschaft eine Begründung für den „Sprung“ in der Größe dieser beiden Kegel ist [48, Sektion 3].

Betrachten wir damit den Graphen der positiv definiten und nichtnegativen Matrix [14, Beispiel 2.9].

$$B = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 & 6 \end{pmatrix}$$


Dann sehen wir, dass dieser Graph einen ungeraden Kreis der Länge 5 enthält. Damit erhalten wir, dass $\mathcal{CP}^5 \subset \mathcal{S}_{\geq 0}^5 \cap \mathcal{N}^5$ ist. Indem wir B in Blockmatrizen der Form $\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix}$ einbetten, erhalten wir solche Graphen für beliebige $d \geq 5$.

Der „Grenzfall“ $d = 5$ für \mathcal{CP}^5 wird genauer in [33] untersucht, so wissen wir beispielsweise, dass die Extremalstrahlen von \mathcal{CP}^5 nur Rang-1-Matrizen sind, unter den Extremalstrahlen von $\mathcal{S}_{\geq 0}^5 \cap \mathcal{N}^5$ treten allerdings auch Rang-3-Matrizen auf [33, Theorem 1]. Dies liefert interessanterweise Faktorisierungsmöglichkeiten, da wir dadurch $\mathcal{S}_{\geq 0}^5 \cap \mathcal{N}^5 = \mathcal{CP}^5 + \text{cone}\{Q \mid Q \text{ ist ein Rang-3-Extremalstrahl von } \mathcal{S}_{\geq 0}^5 \cap \mathcal{N}^5\}$ erhalten.

2.2.4 Zusätzliche Definitionen

Die Menge $[x, y] = \{\lambda x + (1 - \lambda)y \mid \lambda \in [0, 1]\}$ nennen wir *Strecke zwischen x und y* .

Eine *Seite einer abgeschlossenen konvexen Menge K* ist eine echt nichtleere Teilmenge F , die „abgeschlossen“ bezüglich Strecken ist [41, Definition 8.1]. Das heißt, dass eine Strecke $[x, y]$ von Elementen $xy \in K$ F nur dann schneidet, wenn x und y selber in F sind.

F heißt dann auch eine $\dim F$ -Seite von K . Die 0-, 1- und $d-1$ -Seiten von K bezeichnen wir mit *Ecke*, *Kante* und *Facette*. Man beachte, dass die Extremalstrahlen eines Kegels gerade die 1-Seiten sind. In Sektion 2.4 werden wir noch eine Unterklasse von Seiten, sogenannte *offenliegende Seiten* verwenden. Dabei heißt F *offenliegende Seite von K* , falls es eine Hyperebene $H(a, \beta)$ gibt, sodass $F = H(a, \beta) \cap K$ und $K \subseteq H^{\leq}(a, \beta)$ gilt [41, Definition 8.2]. In Sektion 2.4 benötigen die Eigenschaft, dass für volldimensionales K jede Seite von K in einer offenliegenden Seite enthalten ist [41, Theorem 8.4].

Wir bezeichnen die *konvexe Hülle* Δ der Punkte v_1, \dots, v_n

$$\Delta = \text{conv}\{v_1, \dots, v_n\} = \left\{ \sum_{i=1}^n \lambda_i v_i \mid \lambda_i \geq 0, \sum_{i=1}^d \lambda_i = 1 \right\}$$

als *k -dimensionalen Simplex* (kurz *k -Simplex*) mit den Ecken v_1, \dots, v_n , wenn $n = k + 1$ ist und diese Punkte nicht alle in einem $k - 1$ -dimensionalen affinen Unterraum liegen

2.3 Anwendungen

In dieser Sektion wollen wir Anwendungen der kopsitiven und vollständig positiven Matrizen vorstellen. Dabei gehen wir vorwiegend in die Anwendungen im Bereich der Optimierung ein, wo diese Matrizen dazu benutzt werden können, schwierige Probleme als sogenannte konische Optimierungsprobleme darzustellen, und erläutern Vorteile einer solchen Darstellung. Dazu geben wir zunächst eine Einführung in die konische Optimierung und stellen dann die grundsätzlichen Techniken dieser Umformulierungen dar. Für eine ausführliche Behandlung der konischen Optimierung verweisen wir auf [10][81].

Es gibt neben den Anwendungen in der Optimierung eine Vielzahl von weiteren Anwendungen. Da diese den Rahmen und Fokus dieser Arbeit sprengen würde, werden wir auf diese nicht weiter eingehen. Sie finden sich aber in [14][25][27][37][45][48][57][58, Kapitel 10][76][87][90][97][103][104] und Referenzen darin.

2.3.1 Konische Optimierung

Unter einem *konischen Optimierungsproblem*, kurz auch *konisches Problem*, über einem Kegel K verstehen wir ein Problem der Form [10, Sektion 2.2, Gleichung (CP)]

$$\begin{aligned} & \min \langle C, X \rangle, \\ & \text{so dass } \langle A_i, X \rangle = B_i, \quad i = 1, \dots, n \\ & \text{und } X \in K. \end{aligned} \tag{2.8}$$

Die Klasse der konischen Probleme ist recht groß, so lassen sich beispielsweise alle linearen Optimierungsprobleme durch die Wahl von $K = \mathbb{R}_{\geq 0}^d$ in die Form (2.8) bringen.

Ebenfalls lassen sich eine Vielzahl von Problemen als konische Probleme über dem sogenannten *Lorentz-Kegel*

$$L = \left\{ \begin{pmatrix} x \\ x_{d+1} \end{pmatrix} \in \mathbb{R}^{d+1} \mid |x| \leq x_{d+1} \right\} \tag{2.9}$$

schreiben [85]. Solche „second-order-cone-problems“, auch *SOCPs* genannt, können recht effizient in polynomieller Zeit gelöst werden [10, Kapitel 6][96]. Man könnte also annehmen, dass konische Probleme, wie auch die linearen Probleme, „leicht“ sind. Wie wir in der nächsten Sektion sehen werden, ist dies nicht der Fall.

Ein Problem der Form (2.8) heißt *zulässig*, falls der *zulässige Bereich*

$$Z = \{X \mid \langle A_i, X \rangle = B_i, \quad i = 1, \dots, n, X \in K\}$$

nicht leer ist. Man beachte, dass der zulässige Bereich eines konischen Problems für den Fall $n > 0$ geometrisch gerade der Schnitt eines Kegels mit mehreren Hyberebenen ist. Bezeichnen wir mit

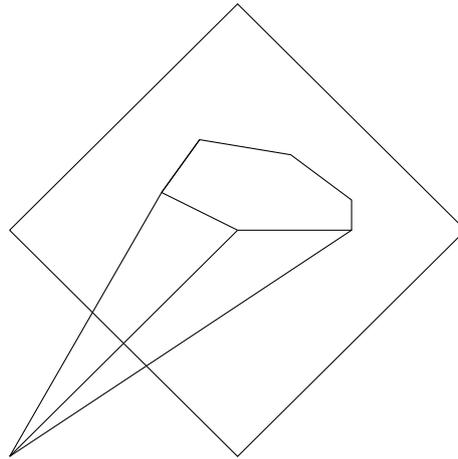
$$L = \{X \mid \langle A_i, X \rangle = 0, \quad i = 1, \dots, n\}$$

und gibt es ein X_0 mit $\langle A_i, X_0 \rangle = B_i$, $i = 1, \dots, n$, so können wir den zulässigen Bereich schreiben als $Z = K \cap (X_0 + L)$. Man beachte, dass L ein Unterraum ist.

Das Problem (2.8) heißt *strikt zulässig*, falls die Menge

$$Z_s = \{X \mid \langle A_i, X \rangle = B_i, i = 1, \dots, n, X \in \text{int}K\}$$

nicht leer ist.



Ein wichtiges Konzept in der linearen Optimierung für Optimalitätskriterien ist die Dualität solcher Probleme [110, Sektion 7.4]. Dieses Konzept lässt sich durch die Dualität von Kegeln, siehe auch Sektion 2.2.2, auf konische Probleme übertragen. Dabei nennen wir dann ein Problem der Form

$$\begin{aligned} & \max y^T B, \\ \text{so dass } & \sum_{i=1}^n y_i A_i - C = S \\ & \text{und } S \in K^* \end{aligned} \tag{2.10}$$

das zu (2.8) *duale Problem* [81, Definition 3.2.5].

Man beachte, dass durch die Menge $\left\{ \sum_{i=1}^n y_i A_i \mid y \in \mathbb{R}^n \right\}$ gerade das orthogonale Komplement von L ist. Wir können den zulässigen Bereich von (2.10) also schreiben als $K^* \cap (-C + L^\perp)$. Damit ist (2.10) ebenfalls wieder ein konisches Problem. Das primale Problem (2.8) wird dazu passend in der Literatur auch oft über dem zulässigen Bereich $K \cap (B + L)$ geschrieben. Dabei gilt auch, dass das duale Problem von (2.10) gerade (2.8) ist, falls der Kegel K abgeschlossen ist [10, Theorem 2.4.1].

Die Probleme (2.8) und (2.10) stehen nun nicht nur in einem geometrischen Zusammenhang, es lassen sich auch Aussagen bezüglich der optimalen Werte der Probleme treffen. So gilt für alle zulässigen Lösungen X und y , dass $\langle C, X \rangle \leq y^T B$ [10, Proposition 2.3.1]. Insbesondere ist der optimale Wert von (2.10) größer gleich dem von (2.8). Ist weiterhin (2.10) von unten beschränkt und strikt zulässig, so stimmen die optimalen Werte überein [10, Theorem 2.4.1]. Weiter ist ein Paar (X, y) von zulässigen Lösungen

der Probleme (2.8) und (2.10) optimal, wenn $\sum_{i=1}^n y_i (\langle A_i, X \rangle - B_i) = 0$ und $y^T B = \langle C, X \rangle$ ist [10, Theorem 2.4.1].

Da es sich bei konischen Problemen um spezielle konvex-lineare Probleme handelt kann man hier ebenfalls die typischen Optimalitätskriterien wie die Slater- und KKT-Bedingungen [81, Sektion 3.1.2] anwenden. Man beachte, dass daher auch die optimalen Lösungen auf dem Rand des Kegels K liegen. In [50, Theorem 3.3] wurden spezielle KKT-artigen Bedingungen für konische Probleme über \mathcal{CP}^d und \mathcal{COP}^d gefunden.

Da uns im Folgenden vorwiegend konische Probleme über den Kegeln \mathcal{COP}^d , \mathcal{CP}^d und $\mathcal{S}_{\geq 0}^d$ begegnen werden, führen wir hier die Kurzbezeichnung *kopositives* und *vollständig positives Problem* ein.

2.3.2 Umformulierungen bekannter schwerer Optimierungsprobleme

Wir wollen nun einige schwierige Optimierungsprobleme als konische Probleme über \mathcal{CP}^d und \mathcal{COP}^d formulieren. Die wohl bekannteste Umformulierung geht auf Bomze et. al [20] zurück und betrifft das *Standard-Quadratische-Problem*

$$\begin{aligned} & \min Q[x], \\ & \text{so dass } e^T x = 1 \\ & \text{und } x \geq 0. \end{aligned} \tag{2.11}$$

Ziel hierbei ist eine quadratische Form Q über dem *Standardsimplex*

$$\Delta^S = \{x \in \mathbb{R}^d \mid x \geq 0, e^T x = 1\} \tag{2.12}$$

zu minimieren. Man beachte, dass Δ^S ein $d - 1$ -Simplex mit den Ecken e_1, \dots, e_d ist.

(2.11) ist \mathcal{NP} -vollständig, da das bekannte \mathcal{NP} -vollständige „MAXCLIQUE“-Problem [73, Main Theorem] durch ein Standard-Quadratisches-Problem dargestellt werden kann. Genauer zeigten Motzkin und Straus, dass für die Cliquenzahl $\omega(G)$ eines Graphen G mit Adjazenzmatrix A [93, Theorem 9]

$$\begin{aligned} \frac{1}{\omega(G)} &= \min (E - A)[x] \\ &, \text{ so dass } x \in \Delta^S \end{aligned} \tag{2.13}$$

gilt.

Die grundsätzliche Idee der Umformulierung ist nun, statt einen nichtnegativen Vektor x eine vollständig positive Matrix X als Variable zu betrachten. Dabei wird unter

anderem der Ausdruck $Q[x]$ „linearisiert“ und man erhält

$$\begin{aligned} & \min \langle Q, X \rangle, \\ & \text{so dass } \langle E, X \rangle = 1 \\ & \text{und } X \in \mathcal{CP}^d. \end{aligned} \tag{2.14}$$

Es bleibt hierbei zu sehen, dass die optimalen Werte von (2.11) und (2.14) übereinstimmen. Dies ist der Fall, da zunächst die Extrempunkte des zulässigen Bereiches von (2.14) gerade die Rang-1-Matrizen xx^T mit $x \in \Delta^S$ sind. Dies folgt im Wesentlichen daher, dass die Rang-1-Matrizen gerade die Extremalstrahlen von \mathcal{CP}^d sind [20, Lemma 5]. Da wir weiter eine lineare Zielfunktion haben folgt, dass der optimale Wert an einem Extrempunkt angenommen wird. Wir sehen also, dass (2.14) tatsächlich eine vollständig positive Umformulierung von (2.11) ist. Man beachte, dass die „vollständig positive Bedingung“ durch die Nichtnegativitätsbedingung in (2.11) kommt.

Wir sehen weiter, dass die Schwierigkeit von (2.11) in die Bedingung $X \in \mathcal{CP}^d$ „geschoben“ wird [48]. In Kapitel 4 behandeln wir kurz die tatsächliche Schwierigkeit des Überprüfen von $X \in \mathcal{CP}^d$.

Ein Vorteil solcher Umformungen ist, dass die Zielfunktion von (2.11) nicht notwendigerweise konvex sein muss (tatsächlich ist sie nur konvex, wenn $Q \in \mathcal{S}_{\geq 0}^d$ ist), aber (2.14) als konisches Problem immer konvex ist. Die in Sektion 2.3.1 erwähnten Optimalitätskriterien können also auf (2.14) angewandt werden.

Das duale Problem zu (2.14) ist ein kopositives Problem und besitzt die Form

$$\max \lambda, \text{ so dass } Q - \lambda E \in \mathcal{COP}^d.$$

Wenden wir dies auf (2.13) an, so erhalten wir

$$\frac{1}{\omega(G)} = \max \omega, \text{ so dass .}$$

Viele graphentheoretische Probleme lassen sich auf recht ähnliche Weise darstellen. De Klerk und Pasechnik zeigten in [37, Korollar 2.4] beispielsweise, dass sich die Stabilitätszahl $\alpha(G)$ eines Graphen durch

$$\min \alpha, \text{ so dass } \alpha(I + A) - E \in \mathcal{COP}^d$$

berechnen lässt, indem sie zuerst ähnlich wie Motzkin und Strauß zeigten, dass es sich durch ein Standard-Quadratisches-Problem berechnen lässt. Die Stabilitätszahl ist die Kardinalität der größten unabhängigen Menge von G , wobei eine Teilmenge von Knoten unabhängig heißt, falls es keine Kante zwischen ihnen gibt. Man beachte, dass dieses „INDSET“-Problem ähnlich wie das „MAXCLIQUE“-Problem \mathcal{NP} -schwer ist [73, Main Theorem]. Gvozdenović und Laurent zeigten in [57, Sektion 2.5] auf ganz ähnliche Weise, dass auch die Färbungszahl eines Graphen auf diese Weise berechenbar ist.

Das Standard-Quadratische Problem gehört zur Problemklasse der quadratischen Optimierung. Interessanterweise lassen sich recht viele Probleme dieser Klasse mit ähnlicher Strategie als vollständig positive Probleme darstellen. Betrachten wir dazu eine allgemeinere Form von (2.11)

$$\begin{aligned} & \min Q[x] + c^T x, \\ & \text{so dass } Ax = b, \\ & x_i \in \{0, 1\} \text{ für } i \in B \\ & \text{und } x \geq 0. \end{aligned} \tag{2.15}$$

Um hier ebenfalls ein vollständig kopositives Problem zu erhalten, betrachten wir wie vorhin statt x eine vollständig positive Matrix X und bezeichnen mit $A_i = a_i a_i^T$. Damit erhalten wir dann ein Problem der Form [31][Gleichung (C)]

$$\begin{aligned} & \min \langle Q, X \rangle + c^T x, \\ & \text{so dass } Ax = b, \\ & \langle A_i, X \rangle = b_i^2, \quad i = 1, \dots, d, \\ & x_i = X_{ii} \text{ für } i \in B \\ & \text{und } \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \in \mathcal{CP}^{d+1}. \end{aligned} \tag{2.16}$$

Damit dies eine tatsächliche Umformung ist, muss (2.15) zulässig sein, dies kann allerdings immer erreicht werden [31]. Man beachte, dass (2.16) im Vergleich zu (2.15) zwar $\mathcal{O}(d^2)$ Variablen, aber nur doppelt so viele Gleichungen besitzt.

In [31, Korollar 2.5] wurde gezeigt, dass wenn der ursprüngliche zulässige Bereich beschränkt ist, ist

$$\left\{ \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \mid (x, X) \text{ im zulässigen Bereich von (2.16)} \right\}$$

gerade die konvexe Hülle der Menge $\left\{ \begin{pmatrix} x \\ 1 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}^T \mid x \text{ im zulässigen Bereich von (2.15)} \right\}$ und sogar, dass der ursprüngliche zulässige Bereich gerade die Projektion des zulässigen Bereiches von (2.16) ist [31, Korollar 2.4].

Damit erhalten wir, dass die optimalen Werte der beiden Probleme übereinstimmen und wenn (x^*, X^*) eine optimale Lösung von (2.16) ist, dann liegt x^* in der konvexen Hülle von optimalen Lösungen von (2.15) [31, Theorem 2.6]. Man beachte, dass x^* selber nicht unbedingt eine optimale Lösung von (2.15) ist. Man beachte auch, dass der Schnitt des Inneren von \mathcal{CP}^{d+1} mit dem zulässigen Bereich von (2.16) leer ist. Insbesondere ist die Umformulierung daher nicht strikt zulässig. Dafür betrachten wir die Matrix $M = \begin{pmatrix} A & -b \end{pmatrix}$. Dann ist

$$M \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} M^T = bb^T - Ax b^T - b x^T A^T + AXA^T = 0,$$

wenn (x, X) im zulässigen Bereich von (2.16) liegt. Damit ist $\begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix}$ positiv semidefinit aber nicht positiv definit und somit nicht im Inneren von \mathcal{CP}^{d+1} . Tatsächlich ist die Bedingung $M \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} M^T = 0$ und sogar nur $M \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} = 0$ äquivalent zu den obigen linearen Nebenbedingungen [32, Proposition 8.3].

Da strikte Zulässigkeit eine wichtige Eigenschaft für viele Optimalitätskriterien und einer Eigenschaft in der nächsten Sektion ist, wird in [31, Sektion 3.1] eine Möglichkeit präsentiert, wie man unter gewissen Voraussetzungen die Variable x in (2.16) eliminieren kann und damit einen zulässigen Bereich erhält, dessen Schnitt mit dem Inneren von \mathcal{CP}^d nicht leer ist. Dafür muss ein Vektor y existieren, sodass $y^T A \geq 0$ und $y^T b = 1$. Diese Bedingung kann in vielen Fällen erreicht werden. Definieren wir nun für so ein y $\alpha = y^T A$, dann sehen wir zunächst, dass die Bedingung $\alpha^T x = 1$ für alle x im zulässigen Bereich von (2.15) erfüllt ist. Daher kann man diese Bedingung zu (2.15) hinzufügen, ohne das Problem zu verändern. Da wir (2.16) mit der Idee, dass $X = xx^T$ ist, konstruiert haben, ist $X\alpha = x$ für alle (x, X) im zulässigen Bereich von (2.16) erfüllt. Da $\alpha \geq 0$ erhalten wir weiter, dass wenn $X \in \mathcal{CP}^{d+1}$ ist, dann ist die Matrix $\begin{pmatrix} X & X\alpha \\ \alpha^T X & 1 \end{pmatrix}$ vollständig positiv. Damit können wir (2.16) schreiben als [31, Theorem 3.1]

$$\begin{aligned} \min \langle Q, X \rangle + c^T X \alpha, \\ \text{so dass } AX\alpha = b, \\ \langle A_i, X \rangle = b_i^2, \quad i = 1, \dots, n, \\ X[\alpha] = 1, \\ (X\alpha)_i = X_{ii} \text{ für } i \in B \\ \text{und } X \in \mathcal{CP}^d. \end{aligned} \tag{2.17}$$

Man beachte, dass es zwar nicht immer gelten muss, aber (2.17) möglicherweise strikt zulässig ist, wohingegen (2.16), wie oben gezeigt, niemals strikt zulässig ist.

Eine weitere sehr allgemeine Form von quadratischen Problemen sind die *quadratisch beschränkten quadratischen Probleme*, auch *QCQPs* genannt. Ein QCQP ist ein Problem der Form

$$\begin{aligned} \max / \min Q[x] + c^T x, \\ \text{so dass } P_i[x] + a_i^T x \leq b_i, \quad i = 1, \dots, n. \end{aligned} \tag{2.18}$$

Die Probleme (2.18) und (2.15) lassen sich in die Form von (2.18) bringen. Man beachte dafür, dass sich die Bedingung $x \in \{0, 1\}$ als $x = x^2$ schreiben lässt.

QCQPs werden wir noch einmal genauer in Kapitel 5 betrachten. Es ist bisher noch unbekannt, ob sich solche Probleme auch als vollständig positive Probleme darstellen lassen [48]. Allerdings ist es möglich, solche Probleme über dem *Kegel der verallgemeinerten vollständig positiven Matrizen bezüglich eines Kegels K*

$$\mathcal{GCP}(K) = \text{cone}\{xx^T \mid x \in K\} \tag{2.19}$$

darzustellen. Man beachte, dass $\mathcal{GCP}(\mathbb{R}_{\geq 0}^d)$ gerade \mathcal{CP}^d und $\mathcal{GCP}(\mathbb{R}^d)$ gerade $\mathcal{S}_{\geq 0}^d$ ist. In [34] werden QCQPs als konische Probleme über $\mathcal{GCP}(K)$ mit $K = \mathbb{R}_{\geq 0}^{n+1} \times L$ oder $K = \mathbb{R}_{\geq 0}^{n+1} \times \mathcal{S}_{\geq 0}^{d+1}$ dargestellt, wobei L der Lorentz-Kegel aus der vorherigen Sektion ist.

Für weitere Umformungen von Problemen ähnlicher Art und Anwendungen dieser Probleme verweisen wir auf [18][20][24][31][95][105][106] und Referenzen darin.

2.3.3 Polyedrische Approximation von \mathcal{CP}^d und \mathcal{COP}^d

In der folgenden Sektion wollen wir auf polyedrische Approximationen von \mathcal{CP}^d und \mathcal{COP}^d eingehen. Wie wir in der vorherigen Sektion erwähnt haben, ist der Grund für die Schwierigkeit von konischen Optimierungsproblemen über \mathcal{CP}^d und \mathcal{COP}^d die Schwierigkeit der Überprüfung von $X \in \mathcal{CP}^d$ beziehungsweise $X \in \mathcal{COP}^d$. Dies kann allerdings in einem gewissen Sinne umgangen werden. Wir werden im Folgenden aufzeigen, dass sich \mathcal{CP}^d und \mathcal{COP}^d polyedrisch approximieren lassen, es also Folgen von Polyedern P_n gibt, sodass entweder

$$K = \bigcap_{n \in \mathbb{N}} P_n \text{ oder } K = \text{cl} \bigcup_{n \in \mathbb{N}} P_n$$

gilt, wobei K entweder \mathcal{CP}^d und \mathcal{COP}^d . Wenn wir dann die konischen Probleme statt über \mathcal{CP}^d oder \mathcal{COP}^d über den Polyedern P_n betrachten, erhalten wir lineare Optimierungsprobleme, welche sich in polynomieller Zeit lösen lassen. Wie wir gleich sehen werden, lassen sich Optimierungsprobleme über \mathcal{CP}^d oder \mathcal{COP}^d mit genügend Rechenaufwand beliebig gut durch solche Schemata approximieren.

Dafür konstruieren wir nun Folgen von polyedrischen Kegeln P_n , sodass

$$P_0 \subseteq P_1 \subseteq \dots \subseteq \mathcal{CP}^d / \mathcal{COP}^d \text{ oder} \\ P_0 \supseteq P_1 \supseteq P_2 \supseteq \dots \supseteq \mathcal{CP}^d / \mathcal{COP}^d$$

gilt. Diese Folgen werden auch *innere beziehungsweise äußere Approximationen* genannt. Hierbei ist zu beachten, dass man durch eine innere Approximation von \mathcal{CP}^d eine äußere Approximation von \mathcal{COP}^d durch Dualität konstruieren kann und umgekehrt.

Die erste polyedrische Approximation, die wir hier vorstellen wollen, wurde von Yıldırım in [121] konstruiert. Dafür betrachten wir zunächst eine Diskretisierung des Standardsimplizes Δ^S . Man beachte, dass eine Matrix Q kopositiv ist, falls $Q[x] \geq 0$ für alle $x \in \Delta^S$ ist. Sei nun

$$\Delta(n) = \{x \in \Delta \mid (n+2)x \in \mathbb{N}^d\}, \text{ für } n = 0, 1, 2, \dots,$$

sowie $\delta(n) = \bigcup_{i=0}^n \Delta(i)$. $\Delta(n)$ und $\delta(n)$ enthalten nur rationale Punkte, für $\Delta(n)$ sind es beispielsweise genau die Punkte x , sodass $(n+2)$ ein Vielfaches des Hauptnenners der Koeffizienten x_i von x ist. Man beachte, dass es für alle n nur endlich viele solcher

Vektoren in Δ^S gibt. Weiter ist $|\Delta(n)| = \binom{d+n+1}{n+2}$. Nun gilt außerdem

$$\delta(0) \subseteq \delta(1) \subseteq \dots \subseteq \Delta^S$$

Damit definieren wir nun die Folge von Kegeln [121, Gleichung (11)]

$$P_n = \{Q \in \mathcal{S}^d \mid Q[x] \geq 0 \text{ für alle } x \in \delta(n)\}. \quad (2.20)$$

Offenbar gilt $P_0 \supseteq P_1 \supseteq \dots \supseteq \mathcal{COP}^d$. Außerdem sind die Kegel P_n aufgrund der Endlichkeit von $\delta(n)$ polyedrisch, da sie durch die endlich vielen Ungleichungen $Q[x] \geq 0$, $x \in \delta(n)$ bestimmt werden, sodass die Folge P_n tatsächlich eine äußere Approximation von \mathcal{COP}^d ist. Es gilt weiterhin [121, Theorem 2.1], dass

$$\mathcal{COP}^d = \bigcap_{n \in \mathbb{N}} P_n.$$

Damit lässt sich nun auch eine innere Approximation von \mathcal{CP}^d durch die dualen Kegel

$$(P_n)^* = \{Q \in \mathcal{S}^d \mid \langle Q, X \rangle \geq 0 \text{ für alle } X \in P_n\} = \left\{ \sum_{x \in \delta(n)} \lambda_x x x^T \mid \lambda_x \geq 0 \right\}$$

angeben. Hier gilt sogar, dass [121, Theorem 2.2]

$$\text{int } \mathcal{CP}^d \subseteq \bigcup_{n \in \mathbb{N}} (P_n)^* \subseteq \mathcal{CP}^d \text{ ist, und damit } \text{cl } \bigcup_{n \in \mathbb{N}} (P_n)^* = \mathcal{CP}^d.$$

Eine innere Approximation an \mathcal{COP}^d wurde durch de Klerk und Pasechnik in [37] gefunden. Dafür betrachten wir für eine Matrix $Q \in \mathcal{S}^d$ die Polynome

$$p_Q(x) = \sum_{i,j=1}^d q_{ij} x_i^2 x_j^2 \text{ und } p_Q^n = \left(\sum_{i,j=1}^d q_{ij} x_i^2 x_j^2 \right) \left(\sum_{i=1}^d x_i^2 \right)^n.$$

Das Polynom $p_Q(x)$ spielt generell für Approximationen von \mathcal{COP}^d eine Rolle, da gezeigt werden kann [101, Sektion 5.3], dass $Q \in \mathcal{COP}^d$ genau dann wenn $p_Q(x) \geq 0$ für alle $x \in \mathbb{R}^d$. Wir definieren nun [37, Definition 3.1]

$$P_n = \{Q \in \mathcal{S}^d \mid p_Q^n(x) \text{ hat nur nichtnegative Koeffizienten}\}. \quad (2.21)$$

Man beachte, dass $\mathcal{N}^d = P_0$ und es gilt [37, Theorem 3.3], dass

$$\mathcal{N}^d = P_0 \subseteq P_1 \subseteq \dots \subseteq \mathcal{COP}^d \text{ und } \text{int } \mathcal{COP}^d \subseteq \bigcup_{n \in \mathbb{N}} P_n \text{ also } \text{cl } \bigcup_{n \in \mathbb{N}} P_n = \mathcal{COP}^d.$$

P_n ist auch hier wieder polyedrisch, sodass wir über diesen Kegeln wieder lineare Optimierung betreiben können. Über Dualität erhalten wir dann eine äußere Approximation von \mathcal{CP}^d .

Weitere Approximationen wurden von Lasserre [80], der über Momente eine äußere

Approximation konstruiert hat, Parillo [101], der sogenannte Summe-von-Quadraten-Faktorisierungen von Polynomen für innere Approximationen verwendet hat, oder Peña, Vera und Zuluaga [102], die ebenfalls Polynom-Bedingungen für eine innere Approximation benutzt haben, gefunden. Einen guten Überblick über verschiedene Approximationen liefert [21, Tabelle 1].

Betrachten wir nun einmal das konische Problem

$$\begin{aligned} & \min \langle C, X \rangle, \\ \text{so dass } & \langle A_i, X \rangle = b_i, i = 1, \dots, n \\ & \text{und } X \in \mathcal{CP}^d. \end{aligned} \tag{2.22}$$

und dessen duales Problem

$$\begin{aligned} & \max b^T y, \\ \text{so dass } & \sum_{i=1}^n y_i A_i + Q = C \\ & \text{und } Q \in \mathcal{COP}^d \end{aligned} \tag{2.23}$$

Sei weiter λ^* der optimale Wert von (2.22) und (2.23), K_n die oben konstruierte innere Approximation von \mathcal{CP}^d und C_n die oben konstruierte äußere Approximation. Man beachte, dass

$$K_0 \subseteq K_1 \subseteq \dots \subseteq \mathcal{CP}^d \subseteq \dots \subseteq C_1 \subseteq C_0 = \mathcal{N}^d,$$

gilt. Daher bezeichnen wir mit

$$\begin{aligned} \lambda_l^n &= \min \{ \langle C, X \rangle \mid \langle A_i, X \rangle = b_i \text{ und } X \in C_n \} \\ \lambda_u^n &= \min \{ \langle C, X \rangle \mid \langle A_i, X \rangle = b_i \text{ und } X \in K_n \}. \end{aligned}$$

die optimalen Werte der über K_n beziehungsweise C_n gelösten linearen Probleme. Damit gilt, dass [121, Gleichung (28)]

$$\lambda_l^0 \leq \lambda_l^1 \leq \dots \leq \lambda^* \leq \dots \leq \lambda_u^1 \leq \lambda_u^0.$$

Weiter erhalten wir unter der Voraussetzung, dass (2.22) und (2.23) beide strikt zulässig sind, dass [121, Theorem 3.1] $\lim_{n \rightarrow \infty} \lambda_l^n = \lim_{n \rightarrow \infty} \lambda_u^n = \lambda^*$. Anders gesagt, dass $\lambda_u^n - \lambda_l^n \rightarrow 0$.

Wir können also die Probleme über \mathcal{CP}^d und \mathcal{COP}^d tatsächlich beliebig genau durch lineare Probleme approximieren. Für eine detailliertere Analyse dieses Sachverhaltes verweisen wir auf [121]. Man beachte, dass die hier vorgestellten und in [21, Tabelle 1] aufgeführten Approximationen regelmäßig sind und daher keine Informationen eines etwaigen Problems benutzen. Es ist allerdings möglich, solche Approximationen zu konstruieren. So haben Bundfuss und Dür in [30] einen auf sogenannten Simplexpartitionen basierenden adaptiven Algorithmus zum linearen Approximieren konischer Probleme über \mathcal{COP}^d an. Auch werden in [57] Approximationen für einige, der in Sektion 2.3.2 angesprochenen, graphentheoretischen Probleme untersucht.

2.4 Die Extremalstrahlen von \mathcal{COP}^d

In Sektion 2.2.1 haben wir schon einen Teil von kositiven Extremalstrahlen aufgeföhrt, die Matrizen E_{ij} und xx^T mit $x \in \mathbb{R}^d \setminus (\mathbb{R}_{\geq 0}^d \cup -\mathbb{R}_{\geq 0}^d)$. Wie sich herausstellt sind dies bereits alle Extremalstrahlen von \mathcal{COP}^d , die in $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ sind. Insbesondere sind dies nach Sektion 2.2.3 alle Extremalstrahlen von \mathcal{COP}^d für $d \leq 4$ [41, Theorem 8.20]. Man beachte hierbei, dass ein Extremalstrahl aus $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ automatisch in $\mathcal{S}_{\geq 0}^d \cup \mathcal{N}^d$ liegt, also nur die Extremalstrahlen dieser beiden Mengen zu betrachten sind.

Wenn $d \geq 5$ ist existieren Extremalstrahlen, welche nicht in $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ liegen, wie wir bereits mit der Horn-Matrix (2.6) gesehen haben. Die Bestimmung aller solcher Extremalstrahlen ist noch ein offenes Problem. Allerdings konnten erst kürzlich in [63] und [1] die 5×5 und 6×6 Extremalstrahlen klassifiziert werden. Für $d \geq 7$ bedarf es noch weiterer Untersuchungen der bisher genutzten Hilfsmittel oder neuer Werkzeuge.

Im Folgenden werden wir die wichtigsten Hilfsmittel zur Klassifikation der nicht positiv semidefiniten oder nicht nichtnegativen Extremalstrahlen von \mathcal{COP}^d vorstellen. Weiter gehen wir auch auf einen Zusammenhang der Extremalstrahlen von \mathcal{COP}^d mit dem Rand von \mathcal{CP}^d ein.

2.4.1 Minimale Nullen einer kositiven Matrix

Das Hauptwerkzeug, welches für die Klassifikation der 5×5 und 6×6 Extremalstrahlen verwendet wurde, ist das Konzept der minimalen Nullen einer kositiven Matrix Q . Dies stellen wir in dieser Sektion vor, dabei folgen wir der Beschreibung von [64].

Dabei heißt ein nichtnegativer Vektor $x \neq 0$ *Null einer kositiven Matrix* Q , falls $Q[x] = 0$. Weiter bezeichnen wir mit $V(Q) = \{x \neq 0 \mid Q[x] = 0\}$ die *Menge der Nullen von* Q .

Wir nennen eine solche Null x *minimal*, wenn die Menge $\text{supp}(x) = \{i \mid x_i \neq 0\}$ minimal ist. Das heißt, wenn es kein $y \in V(Q)$ gibt mit $\text{supp}(y) \subset \text{supp}(x)$. Die *Menge aller minimalen Nullen* wird im Folgenden mit $V_{\min}(Q)$ bezeichnet. Auch schreiben wir $\text{supp}(V(Q)) = \{\text{supp}(x) \mid x \in V(Q)\}$ und analog für $V_{\min}(Q)$.

Der Begriff „minimal“ rechtfertigt sich dadurch, dass wir jede Null von Q als eine endliche Summe der minimalen Nullen schreiben können [64, Korollar 3.4], sie also eine Art Basis von $V(Q)$ bilden.

Das andere Konzept, welches für die Klassifikation benötigt wurde, ist das einer irreduziblen kositiven Matrix. Dabei heißt eine kositive Matrix Q *irreduzibel bezüglich der Menge* M , falls es kein $\varepsilon > 0$ und $B \in M \setminus \{0\}$ gibt, sodass $Q - \varepsilon B \in \mathcal{COP}^d$ ist. Man beachte, dass jeder Extremalstrahl R von \mathcal{COP}^d , der nicht in $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ liegt, notwendigerweise irreduzibel bezüglich dieser beiden Mengen ist.

Daher gilt, dass $\langle V_{\min}(R) \rangle = \mathbb{R}^d$ [64, Theorem 4.5] und für alle Paare $i, j = 1, \dots, d$ gibt es ein $x \in V_{\min}(R)$, sodass $(Rx)_i = (Rx)_j = 0$ und $x_i + x_j > 0$ [64, Korollar 4.2].

Die grundlegende Idee für die Klassifizierung ist es, notwendige Bedingungen für die Menge $\mathcal{I} = \{I_1, \dots, I_m\}$ von Teilmengen der Menge $\{1, \dots, d\}$, sodass $\text{supp}V_{\min}(R)$,

einer bezüglich $\mathcal{S}_{\geq 0}^d$ und \mathcal{N}^d irreduziblen Matrix R mit Einsen auf der Diagonale, durch \mathcal{I} repräsentiert wird, anzugeben. Dabei wurde in [9] gezeigt, dass es ausreicht, Extremalstrahlen mit Einsen auf der Diagonale zu betrachten, dies folgt wesentlich durch (2.1), wie wir in der nächsten Sektion und in Sektion 3.2.3 aufzeigen werden.

Damit lässt sich nun das Hauptresultat für die Klassifizierung von 5×5 und 6×6 Extremalstrahlen vorstellen.

Theorem 2.1. ([64, Theorem 5.7]) *Für eine kopositive, bezüglich $\mathcal{S}_{\geq 0}^d$ und \mathcal{N}^d irreduzible, Matrix Q mit Einsen auf der Diagonale erfüllt die nach Kardinalität geordnete Menge $\mathcal{I} = \{I_1, \dots, I_n\} = \text{supp}V_{\min}(Q)$ folgende Eigenschaften:*

1. $2 \leq I_i \leq d - 2$ für alle $i = 1, \dots, n$.
2. Es gibt keine i, j , so dass $I_j \subset I_i$.
3. Für alle $I \subseteq \{1, \dots, d\}$ und Indizes i, i_1, \dots, i_m, j , sodass
 - $I \subset I_i$
 - $I_{i_k} \setminus I = \{k\}$ für alle $k = 1, \dots, m$
 - $(I_{i_k} \cap I) \subseteq (I_{i_{k+1}} \cap I)$ für alle $k = 1, \dots, m - 1$
 - $I_j \subseteq I \cup \{1, \dots, m\}$

gibt es ein $l \in \{1, \dots, m\}$, sodass $j = i^l$.

4. Sei $m = |\{i \in \{1, \dots, n\} \mid |I_i| = 2\}|$. Seien $G_2(\mathcal{I}) = (V(2), E(2))$, $G_{>2}(\mathcal{I}) = (V(>2), E(>2))$ die Graphen mit $V(2) = \{1, \dots, d\}$, $E(2) = \{I_1, \dots, I_m\}$ und $V(>2) = \{1, \dots, d\} \cup \{m+1, \dots, n\}$, $E(>2) = \{(v, k) \mid v \in I_k\}$. Bezeichnen weiterhin $G_{2,1}, \dots, G_{2,l}$ die bipartiten Zusammenhangskomponenten von G_2 . Dann gibt es Kanten $(v_1, k_1), \dots, (v_l, k_l)$ in $G_{>2}(\mathcal{I})$, sodass v_j ein Knoten von $G_{2,j}$ ist und k_1, \dots, k_l paarweise verschieden sind

5. Für $\alpha_{ij} = \alpha_{ji} \in [0, 1]$ mit $q_{ij} = -\cos(\alpha_{ij}\pi)$ und B mit $b_{ij} = 2\alpha_{ij} - 1$ hat das folgende System eine Lösung
 - Falls $\{i, j\} \in \text{supp}V_{\min}(Q)$, dann ist $\alpha_{ij} = 0$ und $\alpha_{ik} + \alpha_{jk} \geq 1$ für alle k . Ist $\{i, j\} \notin \text{supp}V_{\min}(Q)$, so ist $\alpha_{ij} > 0$
 - Für $\{i, j, k\} \in \text{supp}V_{\min}(Q)$ ist $\alpha_{ij} + \alpha_{ik} + \alpha_{jk} = 1$. Falls es kein $I \in \text{supp}V_{\min}(Q)$ mit $I \subseteq \{i, j, k\}$ gibt, so ist $\alpha_{ij} + \alpha_{ik} + \alpha_{jk} > 1$.
 - Für $i_1, \dots, i_5 \in \{1, \dots, d\}$ ist $\sum a_{i_j i_k} \geq 4$
 - Für $I \in \text{supp}V_{\min}(Q)$ ist $B_I \in \text{conv}\{xx^T \mid x_i \in \{\pm 1\}\}$. Gibt es ein $I \subset J \in \text{supp}V_{\min}(Q)$, dann ist $B_I \in \text{relint}(\text{conv}\{xx^T \mid x_i \in \{\pm 1\}\})$

Man beachte, dass die mit Theorem 2.1 erhaltenen Mengen durch notwendige Bedingungen an die Extremalität von kopositiven Matrizen entstanden sind, aber nicht notwendigerweise zu einem Extremstrahl von \mathcal{COP}^d oder gar einer kopositiven Matrix

gehören müssen, siehe auch [1].

Die durch Theorem 2.1 erhaltenen Mengen kann man weiter durch Einführung einer Äquivalenzrelation verringern. Dabei nennen wir $\mathcal{I} = \{I_1, \dots, I_n\}$ und $\mathcal{J} = \{J_1, \dots, J_n\}$ *äquivalent*, falls es eine Permutation π gibt, sodass $\pi(\mathcal{I}) = \{\pi(I_1), \dots, \pi(I_n)\} = \{J_1, \dots, J_n\} = \mathcal{J}$. Dies ist eine recht natürliche Äquivalenzrelation, wie wir in der nächsten Sektion sehen werden.

Wendet man nun Theorem 2.1 auf alle möglichen Äquivalenzklassen von Mengen $\mathcal{I} \subseteq \mathcal{P}(\{1, \dots, d\})$ an, so reduziert sich die Anzahl dieser wie in Tabelle 2.1.

Erfüllt die Bedingungen	$d = 4$	$d = 5$	$d = 6$	$d = 7$
1,2	10	150	15933	> 14028724
1,2,4,5	6	33	298	19807
1,2,3,5	0	11	2697	> 157872
1,2,3,4	0	2	80	18676
1,2,3,4,5	0	2	44	12378

Tabelle 2.1: Anzahl der Äquivalenzklassen von nichtleeren Mengen \mathcal{I} , die bestimmte Bedingungen von Theorem 2.1 erfüllen, siehe [64, Tabelle 2].

Wie man Tabelle 2.1 ebenfalls entnehmen kann, reicht Theorem 2.1 nicht aus, um auch die 7×7 Extremalstrahlen greifbar zu machen. Man beachte, dass diese Äquivalenzklassen nur zu Extremalstrahlen außerhalb von $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ gehören können. Damit sieht man unter anderem auch, dass es für $d \leq 4$ tatsächlich keine solche Extremalstrahlen gibt. Weiter werden wir in der nächsten Sektion sehen, dass es auch genau 2 Orbits von solchen 5×5 Extremalstrahlen gibt. Für den 6×6 -Fall hat sich herausgestellt, dass es nur 19 Äquivalenzklassen gibt. Daher lässt sich Theorem 2.1 möglicherweise verschärfen, was zur Klassifizierung der 7×7 Extremalstrahlen führen könnte.

2.4.2 Bestimmung der Extremalstrahlen von \mathcal{COP}^d

Bevor wir beschreiben, wie man die Extremalstrahlen von \mathcal{COP}^d aus den in der vorherigen Sektion bestimmten Mengen \mathcal{I} bestimmt, wollen wir einige diesbezüglich wichtige Eigenschaften der kopositiven Extremalstrahlen einführen.

Mit (2.1) wissen wir, dass die kopositiven Extremalstrahlen unter den Abbildungen der Form $(DP)^T X DP$ erhalten bleiben. Die Matrizen der Form DP bilden eine Gruppe G , die auf der Menge der Extremalstrahlen von \mathcal{COP}^d wie oben operiert. Bei der Klassifikation der Extremalstrahlen werden daher verschiedene Klassen von Orbits von Extremalstrahlen bestimmt. Beispielsweise liegen die Extremalstrahlen E_{ij} , $i \neq j$ auf einem Orbit, die Extremalstrahlen E_{ii} auf einem Orbit und die Extremalstrahlen xx^T auf einem Orbit, sodass es bis auf Symmetrie nur 3 Extremalstrahlen aus $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ gibt.

Man beachte, dass zwei Matrizen, die durch äquivalente Mengen \mathcal{I}, \mathcal{J} gegeben werden, auf einem Orbit liegen.

Um nun aus den in der vorherigen Sektion gewonnen Mengen \mathcal{I} von minimalen Nullen Matrizen zu erhalten, benutzt man Bedingung 5 von Theorem 2.1 um wie folgt 2×2 und 3×3 -Blöcke solcher Matrizen zu erhalten. Zunächst wissen wir, dass die Diagonalelemente von Q 1 sind. Ist die Menge $\{i, j\}$ in \mathcal{I} so erhalten wir, dass $q_{ij} = -1$. Ist zusätzlich die Menge $\{j, k\}$ in \mathcal{I} , wobei $i \neq k$ ist, so ist $q_{ik} = -q_{ij} = -q_{jk}$ [1, Lemma 3.1, 3.2]. Für Mengen $\{i, j, k\}$ in \mathcal{I} erhalten wir einen Block der Form

$$\begin{pmatrix} 1 & -\cos(\phi_k) & -\cos(\phi_j) \\ -\cos(\phi_k) & 1 & -\cos(\phi_i) \\ -\cos(\phi_j) & -\cos(\phi_i) & 1 \end{pmatrix},$$

wobei mit Theorem 2.1 gilt, dass $\phi_i, \phi_j, \phi_k \in (0, \pi)$ und $\phi_i + \phi_j + \phi_k = \pi$ ist [1, Lemma 3.2].

Diese „Parametrisierung“ der Matrizen funktioniert, solange \mathcal{I} nur Mengen mit Kardinalität 2 oder 3 enthält. Man beachte, dass dies im 5×5 -Fall immer gilt. Im 6×6 -Fall können solche Mengen \mathcal{I} allerdings auch Mengen mit Kardinalität 4 enthalten. Hierfür wird eine spezielle Behandlung benötigt, welche von den Autoren von [1] durchgeführt wurde, aber recht ähnlich zu der obigen Strategie funktioniert, siehe auch [1, Lemma 4.4]

Mit der obigen Methodik ergeben sich für die 5×5 Extremalstrahlen außerhalb von $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ die Orbits der Matrizen

$$\begin{pmatrix} 1 & -\cos(\phi_4) & \cos(\phi_4 + \phi_5) & \cos(\phi_2 + \phi_3) & -\cos(\phi_3) \\ -\cos(\phi_4) & 1 & -\cos(\phi_5) & \cos(\phi_5 + \phi_1) & \cos(\phi_3 + \phi_4) \\ \cos(\phi_4 + \phi_5) & -\cos(\phi_5) & 1 & -\cos(\phi_1) & \cos(\phi_1 + \phi_2) \\ \cos(\phi_2 + \phi_3) & \cos(\phi_5 + \phi_1) & -\cos(\phi_1) & 1 & -\cos(\phi_2) \\ -\cos(\phi_3) & \cos(\phi_3 + \phi_4) & \cos(\phi_1 + \phi_2) & -\cos(\phi_2) & 1 \end{pmatrix}, \quad (2.24)$$

wobei $\phi_i > 0$ und $\phi_1 + \dots + \phi_5 < \pi$, sowie der Orbit der schon bekannten Horn-Matrix (2.6) [63, Theorem 3.1].

Für den 6×6 -Fall entnehmen wir Tabelle 2.1, dass es, bis auf Äquivalenz, 44 mögliche Mengen \mathcal{I} für Extremalstrahlen von \mathcal{COP}^6 gibt. Diese werden in [64, Tabelle 1] aufgeführt und wurden in [1] untersucht. Dabei blieben 19 übrig, siehe Tabelle 2.2, aus welchen die Orbits der 6×6 Extremalstrahlen bestimmt wurden [1, Theorem 5.1], welche nicht in $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$ liegen oder durch einen 5×5 Extremalstrahl mit (2.7) erzeugt werden.

2.4.3 Maximale Seiten von \mathcal{CP}^d

Wie wir Eingangs erwähnt haben, kann uns die Klassifizierung aller Extremalstrahlen von \mathcal{COP}^d eine Beschreibung des Randes von \mathcal{CP}^d durch Hyperebenen geben.

Dafür betrachten wir zunächst das Konzept einer maximalen Seite. Eine Seite F einer volldimensionalen konvexen Menge K heißt *maximal*, falls sie in keiner anderen Seite enthalten ist [41, Definition 8.5]. Nun bildet die Vereinigung aller maximalen Seiten von

$\text{supp}V_{\min}(R)$
$\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 5\}, \{4, 5\}, \{3, 6\}, \{5, 6\}$
$\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 5\}, \{3, 6\}, \{4, 5, 6\}$
$\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 5\}, \{3, 5, 6\}, \{4, 5, 6\}$
$\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 5, 6\}, \{3, 5, 6\}, \{4, 5, 6\}$
$\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4, 5\}, \{1, 5, 6\}, \{4, 5, 6\}$
$\{1, 2\}, \{1, 3\}, \{1, 4, 5\}, \{2, 4, 6\}, \{3, 4, 6\}, \{4, 5, 6\}$
$\{1, 2\}, \{1, 3\}, \{2, 4, 5\}, \{3, 4, 5\}, \{2, 4, 6\}, \{3, 5, 6\}$
$\{1, 2\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 4, 6\}, \{3, 4, 6\}, \{2, 5, 6\}$
$\{1, 2\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 4, 6\}, \{3, 4, 6\}, \{4, 5, 6\}$
$\{1, 2\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 4, 6\}, \{3, 5, 6\}, \{4, 5, 6\}$
$\{1, 2\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 4, 6\}, \{2, 5, 6\}, \{3, 5, 6\}, \{4, 5, 6\}$
$\{1, 5\}, \{2, 6\}, \{1, 2, 3\}, \{2, 3, 4\}, \{3, 4, 5\}, \{4, 5, 6\}$
$\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 6\}, \{2, 4, 6\}, \{3, 4, 6\}$
$\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 6\}, \{2, 4, 6\}, \{3, 5, 6\}$
$\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 6\}, \{2, 4, 6\}, \{3, 4, 6\}, \{3, 5, 6\}$
$\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 6\}, \{2, 4, 6\}, \{3, 5, 6\}, \{4, 5, 6\}$
$\{1, 2, 3\}, \{2, 3, 4\}, \{3, 4, 5\}, \{4, 5, 6\}, \{1, 5, 6\}, \{1, 2, 6\}$
$\{1, 2, 3\}, \{2, 3, 4\}, \{3, 4, 5\}, \{1, 4, 5\}, \{1, 2, 5\}, \{3, 4, 6\}, \{1, 4, 6\}, \{1, 2, 6\}$
$\{3, 4, 5\}, \{1, 4, 5\}, \{1, 2, 5\}, \{1, 2, 3\}, \{1, 5, 6\}, \{2, 3, 4, 6\}$

Tabelle 2.2: Repräsentanten der Äquivalenzklassen von $\text{supp}V_{\min}(R)$ der nicht positiv semidefiniten und nicht nichtnegativen Extremalstrahlen von \mathcal{COP}^6 , siehe [1, Tabelle 1]

K gerade den Rand von K [41, Theorem 8.8], genauer enthält sogar jede Vereinigung von Seiten, die dem Rand von K entsprechen, alle maximalen Seiten.

Wie in Sektion 2.2.4 angemerkt, ist jede Seite in einer offenliegenden Seite enthalten, sodass maximale Seiten selber offenliegend sein müssen. Ist K nun ein ordentlicher Kegel, dann werden die offenliegenden Seiten von K gerade durch Elemente R von K^* mittels

$$\{Q \in K \mid \langle R, Q \rangle = 0\} = \{Q \mid \langle R, Q \rangle = 0\} \cap K$$

erzeugt [41, Theorem 8.13]. Diese Seite nennt man auch die von R induzierte Seite von K . Die maximalen Seiten in dieser Hinsicht speziell, dass es hier immer einen Extremalstrahl von K^* gibt, der so diese erzeugt [41, Theorem 8.15]. Dies folgt aus den Eigenschaften von Extremalstrahlen.

Würde man also alle Extremstrahlen von \mathcal{COP}^d kennen, so könnte man den Rand von \mathcal{CP}^d klassifizieren. Dies ist allerdings, wie Eingangs erwähnt, ein offenes Problem. Allerdings lassen es die Resultate in [1][41, Theorem 8.20][63] zu, dies wenigstens für $d \leq 6$ zu tun.

Für $d = 2$ erhalten wir daher beispielsweise, dass

$$\text{bd}\mathcal{CP}^2 = \{xx^T \mid x \in \mathbb{R}_{\geq 0}^d\} \cup \{Q \in \mathcal{CP}^2 \mid q_{12} = 0\}.$$

Auch wenn die Klassifizierung aller kpositiven Extremalstrahlen ein offenes Problem ist, so kennen wir bereits zumindest alle Extremalstrahlen aus $\mathcal{S}_{\geq 0}^d + \mathcal{N}^d$. Das ist tatsächlich ausreichend, um alle Facetten von \mathcal{CP}^d zu klassifizieren, wie wir gleich sehen werden. Dies ist ein Resultat von Dickinson [40], welches wir in Kapitel 4 brauchen werden, weshalb wir hier den Beweis angeben wollen. Es gilt das

Lemma 2.2. ([40, Theorem 6.6]) *Die Facetten von \mathcal{CP}^d für $d \geq 2$ werden durch die Extremalstrahlen E_{ij} , $i \neq j$ induziert.*

Beweis. Wir geben hier die Argumentation von Dickinson in [40, Theorem 6.6] an. Zunächst haben die von E_{ij} induzierten Seiten die Form

$$F_{ij} = \{Q \in \mathcal{CP}^d \mid q_{ij} = 0\}.$$

Damit sieht man sofort, dass die Dimension von F_{ij} gerade $\binom{d+1}{2} - 1$ ist, da einerseits die Menge

$$\{xx^T \mid x = e_k + e_l, (k, l) \neq (i, j)\}$$

$\binom{d+1}{2} - 1$ linear unabhängige Elemente des \mathcal{S}^d enthält und selber in F_{ij} enthalten ist. Andererseits ist F_{ij} eine echte Teilmenge von \mathcal{CP}^d . Da \mathcal{CP}^d , wie in Sektion 2.2.1 angemerkt, volldimensional ist, erhalten wir also, dass $\dim F_{ij} = \binom{d+1}{2} - 1$, wenn $i \neq j$ ist. Die Seiten F_{ii} sind in den entsprechenden Seiten F_{ij} enthalten, also dementsprechend keine Facetten. Dies folgt, da zunächst $Q = \sum_k x^k (x^k)^T$ mit $x_k \in \mathbb{R}_{\geq 0}^d$ ist. Falls Q nun in F_{ii} ist, also $q_{ii} = 0$ gilt, muss daher $x_i^k = 0$ für alle k sein. Damit ist dann aber auch $q_{ij} = 0$ für alle j und somit ist Q in F_{ij} .

Um nun zu sehen, dass die F_{ij} tatsächlich die einzigen Facetten sind, betrachten wir die Extremalstrahlen R außerhalb von \mathcal{N}^d . In [40, Lemma 6.4] wurde gezeigt, dass die Dimension einer Seite von \mathcal{CP}^d kleiner oder gleich $\binom{d+1}{2} - |\{i \mid r_{ii} > 0\}|$ ist, wobei R der Extremalstrahl ist, der diese Seite erzeugt. Es bleibt also zu sehen, dass alle Extremalstrahlen außerhalb von \mathcal{N}^d wenigstens zwei positive Diagonalelemente haben.

Da R nicht in \mathcal{N}^d liegt, gibt es ein $r_{ij} < 0$, $i \neq j$. Man beachte hierbei, dass die Diagonalelemente nicht negativ sein können. Daher sind die beiden Diagonalelemente r_{ii} und r_{jj} positiv. Wären beide gleich 0 so wäre $R[e_i + e_j] < 0$, was einen Widerspruch ergeben würde, und wäre nur eines der beiden, angenommen r_{jj} , positiv, so wäre $R[(r_{jj}+1)e_i - r_{ij}e_j] < 0$. Damit können die Seiten, die durch Extremalstrahlen außerhalb von \mathcal{N}^d erzeugt werden, keine Facetten sein. \square

Zwar werden durch die restlichen Extremalstrahlen keine Facetten erzeugt, aber es lassen sich auch hier Aussagen über die Dimension der bekannten positiv semidefiniten und 5×5 -Extremalstrahlen sowie über die Dimensionen der maximalen Seiten von \mathcal{CP}^d machen [41, Theorem 8.33, Theorem 8.35]. Für die 6×6 Extremalstrahlen sind zwar noch keine Resultate beschrieben, allerdings können sie diesbezüglich analog zu den anderen Extremalstrahlen mit den Methoden in [41, Methode 6.5, Theorem 8.34] untersucht werden.

2.5 \mathcal{COP} -perfekte Matrizen

In der folgenden Sektion wollen wir eine für diese Arbeit zentrale Unterklasse von kognitiven Matrizen, die sogenannten \mathcal{COP} -perfekten Matrizen einführen.

Dazu definieren wir zunächst das *kognitive Minimum einer Matrix* Q als [47, Def. 2.1]

$$\min_{\mathcal{COP}} Q = \inf \{Q[v] \mid v \in \mathbb{Z}_{\geq 0}^d \setminus \{0\}\} \quad (2.25)$$

und zugehörig dazu

$$\text{Min}_{\mathcal{COP}} Q = \{v \in \mathbb{Z}_{\geq 0}^d \mid Q[v] = \min_{\mathcal{COP}} Q\} \quad (2.26)$$

die Menge aller nichtnegativen ganzzahligen Vektoren, an denen das kognitive Minimum angenommen wird, welche wir auch als *minimale Vektoren* bezeichnen. Man beachte, dass diese Definitionen ähnlich denen des *arithmetischen Minimums einer Matrix* Q [111, Sektion 1.1.3]

$$\lambda(Q) = \inf \{Q[v] \mid v \in \mathbb{Z}^d \setminus \{0\}\}$$

und der Menge

$$\text{Min}(Q) = \{v \in \mathbb{Z}^d \mid Q[v] = \lambda(Q)\}$$

aller ganzzahligen Vektoren, an denen es angenommen wird, sind. Diese stammen aus der Studie der positiv definiten quadratischen Formen. Wir werden im Folgenden weitere Analogien zwischen dieser Studie und den \mathcal{COP} -perfekten Matrizen sehen.

Für strikt kognitive Matrizen ist das kognitive Minimum positiv und wird an höchstens endlich vielen ganzzahligen Vektoren angenommen [47, Lemma 2.2]. Dies folgt leicht aus einem Kompaktheitsargument. Für Matrizen, die auf dem Rand von \mathcal{COP}^d liegen kann das kognitive Minimum 0 sein und von unendlich vielen Vektoren angenommen werden. Beispielsweise ist das kognitive Minimum der Matrix $Q = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \in \text{bd}\mathcal{COP}^2$ gleich 0 und $\text{Min}_{\mathcal{COP}} Q = \{ke_2 \mid k \in \mathcal{N}\}$. Andererseits ist das kognitive Minimum der Matrix $B = \begin{pmatrix} 2 & -\sqrt{2} \\ -\sqrt{2} & 1 \end{pmatrix} \in \text{bd}\mathcal{COP}^2$ gleich 1, sowie $\text{Min}_{\mathcal{COP}} B = \{e_2\}$. Wir sehen also, dass es für den Rand keine Aussage wie [47, Lemma 2.2] gibt. Man beachte, dass die Matrizen Q und B positiv semidefinit sind, deren kognitives und arithmetisches Minimum übereinstimmen und man damit analoge Aussagen für das arithmetische Minimum treffen kann.

Mit [47, Lemma 2.2] können wir nun die \mathcal{COP} -perfekten Matrizen sinnvoll definieren. Zusätzlich geben wir hier noch die Definition der sogenannten perfekten Matrizen an, welche wir in Kapitel 4 brauchen werden.

Definition 2.3. [47, Definition 2.4], [111, Sektion 3.1.3]

- Ein Matrix $P \in \text{int}\mathcal{COP}^d$ heißt \mathcal{COP} -perfekt, falls es durch sein kognitives

Minimum $\min_{\mathcal{COP}} P$ und $\text{Min}_{\mathcal{COP}} P$ bestimmt wird, also die einzige Lösung des Systems $X[v] = \min_{\mathcal{COP}} P$, $v \in \text{Min}_{\mathcal{COP}} P$ ist.

- Eine Matrix $P \in \mathcal{S}_{>0}^d$ heißt perfekte quadratische Form, falls es durch sein arithmetisches Minimum $\lambda(P)$ und $\text{Min } P$ bestimmt wird, also die einzige Lösung des Systems $X[v] = \min(P)$, $v \in \text{Min } P$ ist.

Eine äquivalente Bedingung für die \mathcal{COP} -Perfektheit einer Matrix P ist, dass der Kegel $\mathcal{V}(P) = \text{cone}\{vv^T \mid v \in \text{Min}_{\mathcal{COP}} P\}$ volldimensional ist [99, Definition 1.2]. Man beachte, dass die Kegel $\mathcal{V}(P)$ in \mathcal{CP}^d liegen und werden auch *Voronoi-Kegel* genannt [111, Sektion 3.1.3]

Wie man sehen kann, sind auch diese beiden Definitionen sehr ähnlich zueinander. Der wesentliche Unterschied ist hierbei die Menge bezüglich der wir die Minimumsbegriffe definieren. Im Falle des kositiven Minimums betrachten wir die Menge $\mathbb{Z}_{>0}^d$, im Falle des arithmetischen Minimums die Menge \mathbb{Z}^d . Tatsächlich lassen sich solche Definitionen bezüglich vieler (diskreter) Mengen formulieren, siehe dazu auch [99, Definition 1.2], wobei man ebenfalls einige analoge Aussagen erhalten wird.

Dass es solche \mathcal{COP} -perfekten Matrizen für alle $d \geq 2$ gibt wird klar, indem man die perfekte quadratische Form

$$Q_{A_d} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

mit $\min_{\mathcal{COP}} Q_{A_d} = 2$ und $\text{Min}_{\mathcal{COP}} Q_{A_d} = \left\{ \sum_{i=j}^k e_i \mid 1 \leq j \leq k \leq d \right\}$ betrachtet. Man sieht, dass $\mathcal{V}(Q_{A_d})$ volldimensional ist. Nun ist Q_{A_d} positiv definit und damit strikt kositiv, also tatsächlich \mathcal{COP} -perfekt. Für $d = 1$ ist dies trivial.

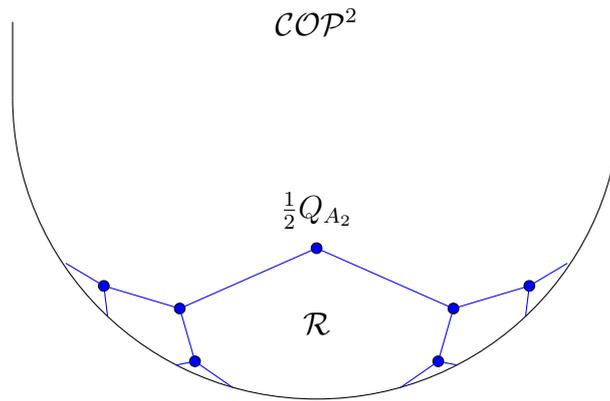
Eine wichtige Menge im Studium der perfekten quadratischen Formen ist das *Ryshkov-Polyeder* [111, Sektion 3.1.1]

$$\mathcal{R}_\lambda = \{Q \in \mathcal{S}_{>0}^d \mid \lambda(Q) \geq \lambda\}. \quad (2.27)$$

Wie auch schon die Male davor definieren wir eine Art „Ryshkov-Polyeder!kositives“ [47, Sektion 2.2]

$$\mathcal{R} = \{Q \in \mathcal{S}^d \mid Q[v] \geq 1 \text{ für alle } v \in \mathbb{Z}_{\geq 0}^d \setminus \{0\}\}. \quad (2.28)$$

Wie man sieht, ist das Ryshkov-Polyeder \mathcal{R}_λ für $\lambda \geq 1$ sogar in \mathcal{R} enthalten. Zwei Eigenschaften, die sich diese Mengen teilen sind zunächst, dass beide sogenannte *lokal endliche Polyeder* sind [46, Lemma 2.4][111, Theorem 3.1]. Das sind Schnitte von


 Abbildung 2.3: Der lokal endliche Polyeder \mathcal{R} für $d = 2$

abzählbar unendlich vielen Halbräumen, so dass jeder Schnitt mit einem beliebigen Polytop wieder ein Polytop ist [111, Sektion 3.1.1]. Die in dieser Arbeit wichtigere Eigenschaft ist aber, dass die perfekten beziehungsweise \mathcal{COP} -perfekten Matrizen P mit $\lambda(P) = \lambda$ beziehungsweise $\min_{\mathcal{COP}} P = 1$ gerade die Ecken der jeweiligen Polyeder sind.

An der Definition des Ryshkov-Polyeders ist zu sehen, dass er im Inneren von $\mathcal{S}_{\geq 0}^d$ liegt. Auch \mathcal{R} liegt im Inneren von \mathcal{COP}^d [46, Lemma 2.3]. Man beachte, dass für jede Matrix Q in \mathcal{R} gilt, dass $\min_{\mathcal{COP}} Q \geq 1$. Damit können wir (2.28) umschreiben zu

$$\mathcal{R} = \{Q \in \text{int } \mathcal{COP}^d \mid \min_{\mathcal{COP}} Q \geq 1\},$$

wobei hier nun besser ersichtlich ist, dass dies tatsächlich eine „kopositive Version“ von (2.27) ist.

Weiter erhalten wir damit eine Darstellung des Inneren von \mathcal{COP}^d . Es ist

$$\text{int } \mathcal{COP}^d = (\text{cone } \mathcal{R}) \setminus \{0\}, \quad (2.29)$$

da jede strikt kopositive Matrix Q wegen [47, Lemma 2.2] auf einem Strahl liegt, der von einer strikt kopositiven Matrix B mit $\min_{\mathcal{COP}} B = 1$ erzeugt wird. (2.29) werden wir in Kapitel 4 für eine Darstellung von \mathcal{CP}^d gebrauchen.

Weitere Eigenschaften der \mathcal{COP} -perfekten Matrizen werden wir in Kapitel 3 und 4 betrachten.

3 Ein Simplexalgorithmus für rationale \mathcal{CP} -Faktorisierung

In diesem Kapitel wollen wir uns mit dem von Dutour Sikirić, Schürmann und Vallentin in [47] entwickeltem Algorithmus zur Bestimmung einer, falls existenten, rationalen \mathcal{CP} -Faktorisierung einer Matrix $Q \in \mathcal{CP}^d$ befassen. Dabei heißt eine Menge $\{x_1, \dots, x_n\}$ nichtnegativer Vektoren und ein Vektor $\alpha \in \mathbb{R}_{\geq 0}^n$ \mathcal{CP} -Faktorisierung einer Matrix Q , wenn $Q = \sum_i \alpha_i x_i x_i^T$ ist. Sind dabei die x_i und α rational, so sprechen wir von einer

Input : $Q \in \widetilde{\mathcal{CP}}^d$

Output: Rationale \mathcal{CP} -Faktorisierung von Q

Wähle eine erste \mathcal{COP} -perfekte Matrix $P = P_{\text{Start}} \in \mathcal{R}$ (2.28)

Initialisiere $\mathcal{V}(P) = \text{cone}\{vv^T \mid v \in \text{Min}_{\mathcal{COP}} P\}$

while $Q \notin \mathcal{V}(P)$ **do**

Bestimme einen Extremalstrahl R von $\mathcal{V}(P)^*$ mit $\langle R, Q \rangle < 0$

Bestimme eine benachbarte \mathcal{COP} -perfekte Matrix $N = P + \lambda R$ mit $\lambda > 0$
und $\min_{\mathcal{COP}} N = 1$

Setze $P = N$

Bestimme $\mathcal{V}(P)$

end

Bestimme $\alpha_1, \dots, \alpha_n \in \mathbb{Q}_{\geq 0}$, sodass $Q = \sum_{i=1}^n \alpha_i v_i v_i^T$ und gebe Faktorisierung aus.

Algorithmus 1: Algorithmus zum Bestimmen einer rationalen \mathcal{CP} -Faktorisierung, siehe [47, Algorithmus 1]

rationalen \mathcal{CP} -Faktorisierung der Matrix Q . Die Menge aller rational faktorisierten vollständig positiven Matrizen bezeichnen wir mit

$$\widetilde{\mathcal{CP}}^d = \left\{ \sum_{i=1}^n \alpha_i x_i x_i^T \mid n \in \mathbb{N}, \alpha_i \in \mathbb{Q}_{\geq 0}, x_i \in \mathbb{Q}_{\geq 0}^d \right\}. \quad (3.1)$$

Wie viele Matrizen tatsächlich rational faktorisiert werden können, ist noch ein offenes Problem. Es besteht allerdings die Vermutung, dass dies auf jede rationale vollständig positive Matrix zutrifft [115, Vermutung 1]. Bisher konnte diesbezüglich neben kleineren Resultaten gezeigt werden, dass diese Vermutung für alle rationalen Matrizen im Inneren von \mathcal{CP}^d gilt [46, Theorem 1.1]. Für Algorithmus 1 ist es wichtig anzumerken, dass wir uns bei rationalen \mathcal{CP} -Faktorisierungen auf ganzzahlige Vektoren x_i beschränken

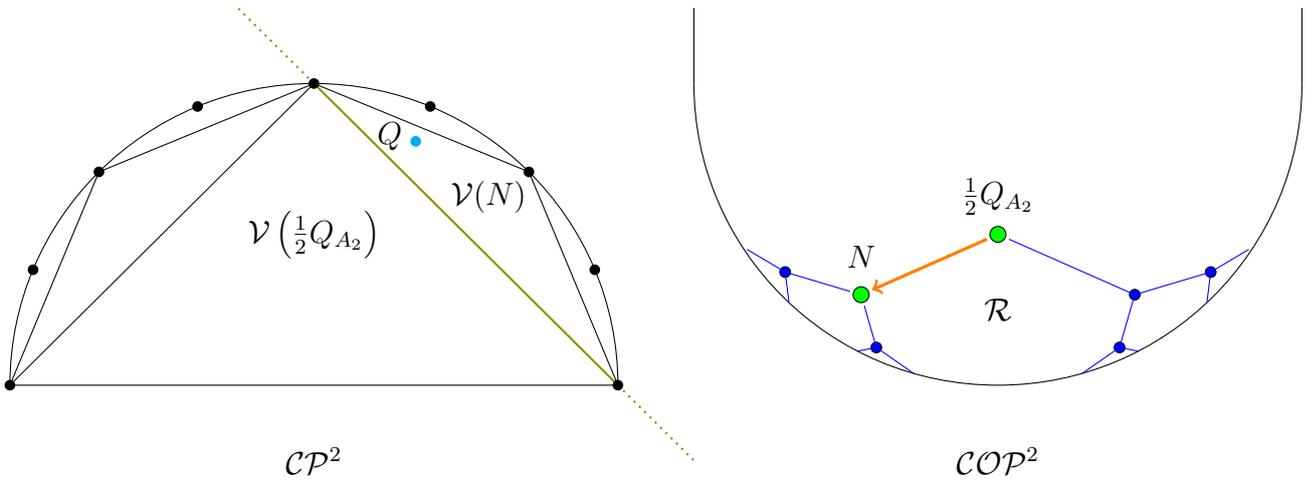


Abbildung 3.1: Darstellung der Funktionsweise von Algorithmus 1. Hier wurde $\frac{1}{2}Q_{A_2}$ als Startmatrix gewählt sowie der Extremalstrahl in Richtung N (orange). Dieser induziert eine Hyperebene (olivfarbend). Da sich die zu faktorisierende Matrix Q in $\mathcal{V}(N)$ befindet, terminiert der Algorithmus hier mit der entsprechenden \mathcal{CP} -Faktorisierung.

können, indem wir die Vektoren x_i mit dem Hauptnenner ihrer Koeffizienten skalieren. Dadurch verändert sich natürlich α . Insbesondere können wir also

$$\widetilde{\mathcal{CP}}^d = \left\{ \sum_{i=1}^n \alpha_i x_i x_i^T \mid n \in \mathbb{N}, \alpha_i \in \mathbb{Q}_{\geq 0}, x_i \in \mathbb{Z}_{\geq 0}^d \right\}$$

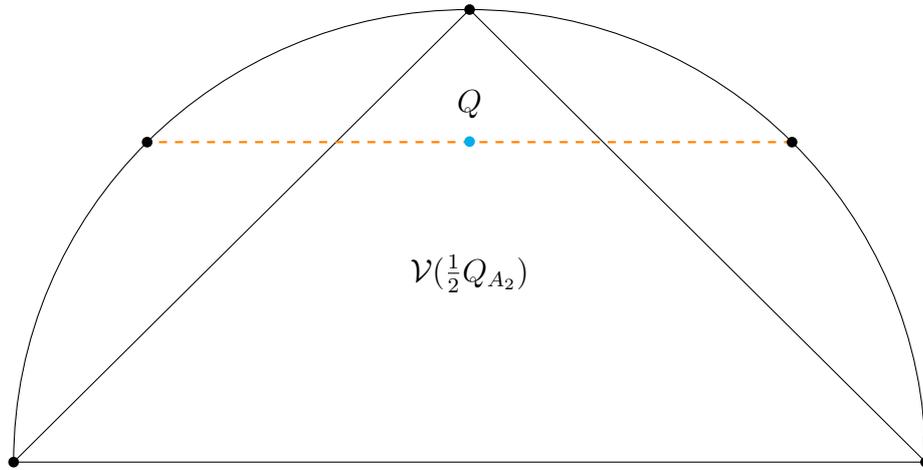
setzen.

Ein Konzept welches mit \mathcal{CP} -Faktorisierungen zusammenhängt, ist der sogenannte \mathcal{CP} -Rang einer Matrix Q . Eine positiv semidefinite Matrix Q kann ähnlich wie vollständig positive Matrizen in eine Summe von Rang-1-Matrizen xx^T zerlegt werden [14, Theorem 1.10]. Diese müssen natürlich in diesem Fall nicht unbedingt nichtnegativ sein. Die minimale Anzahl an Rang-1-Matrizen die benötigt wird, um eine Matrix Q zu zerlegen, entspricht gerade dem Rang von Q [14, Bemerkung 1.1]. Es ist also naheliegend zu überlegen, ob dies auch im Falle von \mathcal{CP} -Faktorisierung gilt. Dabei nennen wir dann die minimale Anzahl an nichtnegativen Rang-1-Matrizen die benötigt wird, um eine vollständig positive Matrix Q zu faktorisieren, den \mathcal{CP} -Rang von Q , auch mit cprk bezeichnet. Man beachte, dass der \mathcal{CP} -Rang einer Matrix immer größer oder gleich dem Rang der Matrix ist.

Wir wollen hier kurz Algorithmus 1 aus dem Standpunkt minimaler Faktorisierungen betrachten, für eine ausführliche Übersicht der Resultate und offenen Probleme bezüglich des \mathcal{CP} -Rangs verweisen wir auf [12][13][14][15][113][114] [115].

Betrachten wir einmal eine rationale Matrix $Q \in \widetilde{\mathcal{CP}}^2$ mit \mathcal{CP} -Rang 2, wie in der unteren Abbildung dargestellt. Q liegt daher in der konvexen Hülle zweier Extremalstrahlen

xx^T von \mathcal{CP}^2 .



Allerdings liegt Q auch im Inneren des Kegels $\mathcal{V}(\frac{1}{2}Q_{A_2})$. Daher wird Algorithmus 1 eine \mathcal{CP} -Faktorisierung mit 3 Rang-1-Matrizen ausgeben, was offensichtlich mehr sind als benötigt werden. Solche Beispiele lassen sich für beliebige $d \geq 2$ konstruieren, sodass Algorithmus 1 im Allgemeinen keine minimale Faktorisierung berechnet. Tatsächlich bestehen die von Algorithmus 1 berechneten \mathcal{CP} -Faktorisierungen im Allgemeinen aus wenigstens $\binom{d+1}{2}$ Vektoren.

In den folgenden Sektionen geben wir zunächst den Beweis der Termination von Algorithmus 1 nach endlich vielen Schritten aus [47] und beschreiben dann detailliert die Funktionsweise.

3.1 Termination des Algorithmus

Wenn wir in der **while**-Schleife den Extremalstrahl R geeignet wählen, dann terminiert Algorithmus 1 nach endlich vielen Schritten. Es gilt also das

Theorem 3.1. ([47, Theorem 3.1]) *Für $Q \in \widetilde{\mathcal{CP}}^d$ terminiert Algorithmus 1 bei geeigneter Wahl von R nach endlich vielen Schritten.*

Man beachte, dass Algorithmus 1 nicht terminieren muss, wenn Q in $\mathcal{CP}^d \setminus \widetilde{\mathcal{CP}}^d$ liegt [47, Sektion 4.1].

Für den Beweis von Theorem 3.1 folgen wir der Argumentation in [47]. Zunächst überlegen wir uns, dass es für jede Matrix Q in $\widetilde{\mathcal{CP}}^d$ immer eine \mathcal{COP} -perfekte Matrix P gibt, sodass $Q \in \mathcal{V}(P)$ ist. Dafür betrachten wir die Mengen

$$D = \{xx^T \mid x \in \mathbb{Z}_{\geq 0}^d \setminus \{0\}\}, D_C = \text{conv}D \subset \mathcal{CP}^d.$$

Ist nun H eine Hyperebene, die $\binom{d+1}{2}$ linear unabhängige Punkte aus D enthält und H^\geq der Halbraum der 0 enthält. Enthält H^\geq keine Punkte aus $D \setminus H$, dann ist die Matrix

P mit $P[x] = 1$ für $x \in H \cap D$ \mathcal{COP} -perfekt [99, Definition 1.2f]. Man erhält also eine Bijektion zwischen \mathcal{COP} -perfekten Matrizen und den Facetten von D_C .

Damit erhalten wir eine Überdeckung von $\widetilde{\mathcal{CP}}^d$ durch

$$\widetilde{\mathcal{CP}}^d \subseteq \bigcup_{P \text{ } \mathcal{COP}\text{-perfekt}} \mathcal{V}(P) \subset \mathcal{CP}^d. \quad (3.2)$$

Dies folgt, indem wir die vollständig positive Matrix $Q = \sum_{i=1}^n \alpha_i x_i x_i^T$, $\alpha_i \in \mathbb{Q}_{\geq 0}$, $x_i \in \mathbb{Z}_{\geq 0}^d$ aus $\widetilde{\mathcal{CP}}^d$ betrachten. Multiplizieren wir Q mit $\mu = \sum_{i=1}^n \alpha_i \geq 0$ so liegt das Resultat in D_C . Also trifft der von Q erzeugte Strahl eine Facette F von D_C . Daher ist $Q \in \text{cone}F = \mathcal{V}(P)$ für eine \mathcal{COP} -perfekte Matrix P .

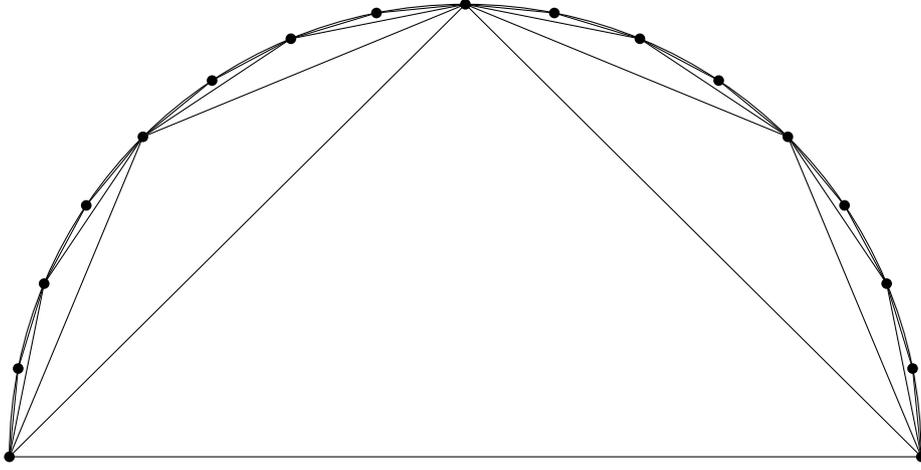


Abbildung 3.2: Unterteilung von $\widetilde{\mathcal{CP}}^2$ in Kegel $\mathcal{V}(P)$ nach (3.2)

Da nun klar ist, dass es für jede Matrix Q in $\widetilde{\mathcal{CP}}^d$ einen Kegel $\mathcal{V}(P)$ gibt, sodass Q in $\mathcal{V}(P)$ liegt, bleibt es noch zu zeigen, dass solch ein Kegel, bei geeigneter Wahl des Extremalstrahles in Algorithmus 1, nach endlich vielen Schritten erreicht wird.

Dafür betrachten wir die Menge

$$\mathcal{P}(Q, \lambda) = \{B \in \mathcal{R} \mid \langle B, Q \rangle \leq \lambda\}.$$

In [46, Lemma 2.4] wurde gezeigt, dass $\mathcal{P}(Q, \lambda)$ für $Q \in \text{int } \mathcal{CP}^d$ und $\lambda > 0$ ein Polytop ist. Man beachte, dass daher $\mathcal{P}(Q, \lambda)$ nur endlich viele Ecken besitzt [8, Korollar 4.3].

Bezeichnet P^* eine \mathcal{COP} -perfekte Matrix, so dass $Q \in \mathcal{V}(P^*)$ ist. Wählen wir λ groß genug, dass P_{Start} und P^* Ecken von $\mathcal{P}(Q, \lambda)$ sind, dann folgt die Behauptung in Theorem 3.1 für $Q \in \text{int } \mathcal{CP}^d \cap \widetilde{\mathcal{CP}}^d$. Denn zunächst nimmt der Ausdruck $\langle P, Q \rangle$ im Verlauf des Algorithmus stets ab, da R so gewählt wird, dass $\langle R, Q \rangle < 0$ ist. Es gilt also für die Matrizen P und N in Algorithmus 1, dass

$$\langle N, Q \rangle = \langle P + \lambda R, Q \rangle < \langle P, Q \rangle \text{ ist.}$$

Liegt also P_{start} in $\mathcal{P}(Q, \lambda)$, so „verlassen“ wir $\mathcal{P}(Q, \lambda)$ im Verlauf des Algorithmus nicht mehr. Da jede \mathcal{COP} -perfekte Matrix in $\mathcal{P}(Q, \lambda)$ eine Ecke von $\mathcal{P}(Q, \lambda)$ ist, erreichen wir also nach höchstens endlich vielen Schritten P^* .

Falls Q auf dem Rand von \mathcal{CP}^d liegt, liegt es mit (3.2) auch auf dem Rand eines Kegels $\mathcal{V}(P)$. Damit gibt es in jeder Umgebung von Q Matrizen B aus $\text{int } \mathcal{CP}^d \cap \widetilde{\mathcal{CP}}^d$ für die die Behauptung in Theorem 3.1 stimmt.

Man beachte, dass für alle $\mathcal{V}(P)$ mit $Q \notin \mathcal{V}(P)$ eine Trennhyperebene H , welche durch einen Extremalstrahl R von $(\mathcal{V}(P))^*$ mit $\langle R, Q \rangle < 0$ gegeben wird, existiert, sodass $Q \in H^<$ ist [110, Theorem 7.1]. Damit existiert insbesondere eine Umgebung U von Q , so dass $U \subset H^<$.

Offenbar gibt es dann auch für jede endliche Anzahl an Trennhyperebenen eine Umgebung von Q , die auf der selben „Seite“ liegt. Da in jeder Umgebung von Q Matrizen aus $\text{int } \mathcal{CP}^d \cap \widetilde{\mathcal{CP}}^d$ liegen, für die Algorithmus 1 nach dem ersten Teil nur endlich viele Schritte braucht, folgt die Behauptung durch einen Widerspruch. \square

Algorithmus 1 findet also, wenn existent, in endlicher Zeit eine rationale \mathcal{CP} -Faktorisierung. Man beachte, dass theoretisch auch bestimmte reelle \mathcal{CP} -Faktorisierungen gefunden werden können, solange die dazugehörige Matrix Q in $\bigcup \mathcal{V}(P)$ liegt.

3.2 Analyse von Algorithmus 1

Wir wollen nun auf die einzelnen Schritte von Algorithmus 1 eingehen, sowie alle Kenntnisse, die für eine Implementation von Nöten sind, vorstellen.

Für eine Startmatrix bietet sich die, in jeder Dimension bekannte, \mathcal{COP} -perfekte Matrix $\frac{1}{2}Q_{A_d}$ an. Andere Wahlen sind durchaus möglich, so sind beispielsweise bestimmte perfekte quadratische Formen ebenfalls \mathcal{COP} -perfekt. Eine Frage, die man sich hier stellen kann, ist die Folgende:

Frage. Ist es möglich aus einer Matrix Q Informationen über eine, im Sinne von Algorithmus 1, geeignete \mathcal{COP} -perfekte Matrix zu gewinnen?

Man beachte, dass sich die Wahl der Startmatrix durchaus auf die Laufzeit von Algorithmus 1 auswirken kann, wie wir in Sektion 4.3 sehen werden.

Die Bedingung der **while**-Schleife lässt sich durch das lineare Optimierungsproblem

$$\min \{ \langle B, Q \rangle \mid B \in (\mathcal{V}(P))^* \} \quad (3.3)$$

überprüfen. Ist $Q \in \mathcal{V}(P)$, so ist nach Definition $\langle B, Q \rangle \geq 0$ für alle $B \in (\mathcal{V}(P))^*$ und somit, da $0 \in (\mathcal{V}(P))^*$, der optimale Wert von (3.3) gleich 0. Ist umgekehrt der optimale Wert von (3.3) gleich 0, so gilt, dass $\langle B, Q \rangle \geq 0$ für alle $B \in (\mathcal{V}(P))^*$ und daher $Q \in \mathcal{V}(P)$.

Sobald die **while**-Schleife verlassen wird, können wir die Koeffizienten der rationalen \mathcal{CP} -Faktorisierung von $A \in \mathcal{V}(P)$ durch das lineare Optimierungsproblem

$$\begin{aligned} & \min 0^T \alpha, \\ & \text{so dass } \begin{pmatrix} \vdots & & \vdots \\ \text{vec}(v_1 v_1^T) & \dots & \text{vec}(v_n v_n^T) \\ \vdots & & \vdots \end{pmatrix} \alpha = \text{vec}(Q), \\ & n = |\text{Min}_{\mathcal{COP}} P|, v_i \in \text{Min}_{\mathcal{COP}} P \text{ und } \alpha \geq 0 \end{aligned}$$

bestimmen. Hierbei ist für eine Matrix $A \in \mathbb{R}^{d \times n}$ $\text{vec}(A)$ der Vektor $m \in \mathbb{R}^{dn}$ der durch „Aufeinanderstaffeln“ der Zeilen von M entsteht.

Man beachte, dass hier aufgrund von Symmetrie $\text{vec}(A)$ ein Vektor der Länge $\binom{d+1}{2}$ ist, wobei die Nichtdiagonaleinträge verdoppelt werden müssen. Man beachte weiter, dass die Existenz von solchen $\alpha_i \in \mathbb{Q}_{\geq 0}$ durch den Satz von Carathéodory [110, Korollar 7.1i] garantiert wird.

3.2.1 Bestimmung eines geeigneten Extremalstrahles von $(\mathcal{V}(P))^*$

Wir wollen nun erklären, wie sich ein geeigneter Extremalstrahl von $(\mathcal{V}(P))^*$ in der **while**-Schleife von Algorithmus 1 bestimmen lässt. Dazu besprechen wir zunächst Methoden zur algorithmischen Berechnung von Extremalstrahlen punktierter polyedrischer Kegel.

Man beachte, dass $(\mathcal{V}(P))^*$ ein polyedrischer Kegel ist, der durch den Schnitt der Halbräume

$$H_x^{\geq} = \{B \in \mathcal{S}^d \mid B[v] = \langle B, vv^T \rangle \geq 0\}, \quad v \in \text{Min}_{\text{COP}} P$$

gegeben wird. Zusätzlich ist $(\mathcal{V}(P))^*$ punktiert, denn mit Definition 2.3 (siehe auch [99, Definition 1.2]) ist

$$\text{rk} \{v_1 v_1^T, \dots, v_n v_n^T\} = \binom{d+1}{2}, \quad v_i \in \text{Min}_{\text{COP}} P, \quad n = |\text{Min}_{\text{COP}} P|.$$

Algorithmische Berechnung der Extremalstrahlen eines polyedrischen punktierten Kegels

Der Einfachheit halber betrachten wir die Methoden sowie die Theorie dahinter im Falle von Kegeln im \mathbb{R}^d statt dem \mathcal{S}^d . Daher wird ein polyedrischer Kegel K gegeben durch

$$K = \{x \in \mathbb{R}^d \mid Ax \geq 0\},$$

für eine Matrix $A \in \mathbb{R}^{n \times d}$. Die Zeilen a_i der Matrix entsprechen dann den Normalenvektoren der Halbräume, deren Schnitt K ist. Man beachte, dass K punktiert ist, falls $\text{rk} A = d$.

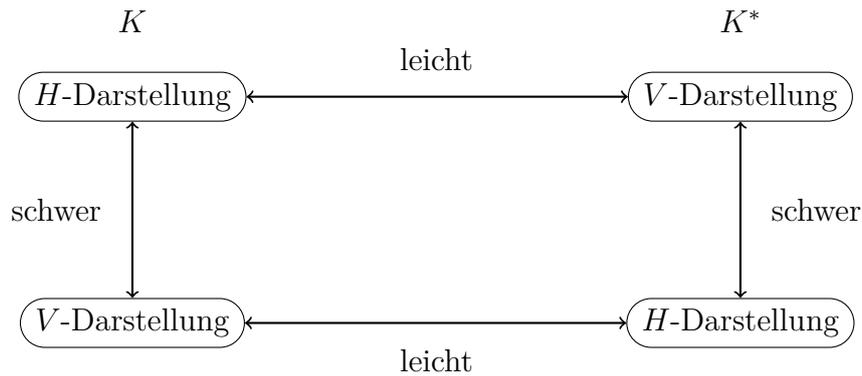
Die Extremalstrahlen von punktierten polyedrischen Kegeln lassen sich theoretisch „einfach“ bestimmen. Dazu suchen wir ein $i^* \in \{1, \dots, n\}$ und eine Teilmenge $I^* \subseteq \{1, \dots, n\} \setminus \{i^*\}$, sodass $\text{rk}(\bigcup_{i \in I^*} \{a_i\} \cup \{a_{i^*}\}) = d$. Dann wird ein Extremalstrahl von K gegeben durch [110, Sektion 8.8, Gleichung (26)]

$$\{v \in K \mid \langle a_{i^*}, v \rangle > 0, \langle a_i, v \rangle = 0 \text{ für alle } i \in I^*\}. \quad (3.4)$$

Man beachte, dass 3.4 durchaus leer sein kann. In diesem Fall wird durch diese Auswahl von Indizes kein Extremalstrahl gegeben. Man sieht, dass dadurch ein polyedrischer Kegel nur endlich viele Extremalstrahlen haben kann.

Wie in Sektion 2.2.1 angemerkt wurde, sind polyedrische Kegel endlich erzeugt. Aus der Definition von Extremalstrahlen, siehe Sektion 2.2.1, folgt daher, dass jede erzeugende Menge von K alle Extremalstrahlen beinhalten muss. Also ist das Problem der Bestimmung aller Extremalstrahlen von K äquivalent zur Bestimmung einer kleinsten erzeugenden Menge von K .

Die Darstellung von K durch ein Ungleichungssystem wird auch H -Repräsentation, die als konische Hülle von endlich vielen Punkten auch V -Repräsentation genannt. Es ist leicht, die jeweils andere Darstellung des dualen Kegels K^* zu berechnen, aber ein schweres Problem aus der H -Darstellung die V -Darstellung von K zu bestimmen und



umgekehrt. Dieser Prozess wird auch Darstellungsumwandlung genannt. Tatsächlich ist das Problem alle Extremalstrahlen zu berechnen, gerade die Darstellungsumwandlung von der H - zur V -Darstellung. Man beachte, dass die hier vorgestellten Methoden auch zur Bestimmung aller Facetten oder äquivalent der H -Darstellung aus der V -Darstellung eines Kegels verwendet werden können, siehe dazu auch [3, Sektion 8.1].

Bevor wir uns mit den praktischen algorithmischen Methoden beschäftigen sei angemerkt, dass es noch ein offenes Problem ist, ob es einen Algorithmus zum Bestimmen der Extremalstrahlen eines Kegels gibt, welcher polynomiell in der Eingabe- und Ausgabegröße ist [86, Kapitel 8]. Man beachte, dass es keinen Algorithmus geben kann, der polynomiell in der Eingabegröße ist, denn die maximale Anzahl der Extremalstrahlen eines d -dimensionalen Kegels, welcher durch n Halbräume gegeben wird, ist gleich $n^{\lfloor \frac{d}{2} \rfloor}$ [88, Upper Bound Theorem], also exponentiell in den Eingabegrößen. Ein Algorithmus der dieses Optimum annimmt, also einen Aufwand von $\mathcal{O}\left(n^{\lfloor \frac{d}{2} \rfloor}\right)$ besitzt, wird in [108] angegeben.

Die zwei wohl bekanntesten Algorithmen zur Bestimmung der Extremalstrahlen sind die von Motzkin et. al. entworfene „double description method“ [92] und der „reverse-search“-Algorithmus von Avis und Fukuda [3]. Natürlich gibt es noch weitaus mehr Methoden, die wir hier zwar nicht erwähnen wollen, jedoch in [54][86], sowie Referenzen darin gefunden werden können. Effiziente Implementationen dieser beiden Algorithmen finden sich in [4] und [7].

Wie wir in den folgenden Sektionen sehen werden, sind diese beiden Methoden sehr unterschiedlich, sodass sie schwer vergleichbar sind. Es scheint sich aber eine Art „Faustregel“ herausgebildet zu haben, welche Methode für welche Kegel angewendet werden soll, siehe dazu auch [86, Seite 147]. Für diese Faustregel definieren wir zuerst einen Kegel K als *entartet*, falls es ein $x \neq 0$ in K gibt, welches mehr als $d + 1$ Ungleichungen mit Gleichheit erfüllt oder äquivalent, welches in mehr als $d + 1$ Facetten liegt [54].

Es hat sich durch Experimente über die Jahre bewährt, dass die double-description Methode verwendet werden sollte, falls K stark entartet ist. Andernfalls, also wenn K nicht oder nur schwach entartet ist, soll der reverse-search Algorithmus besser funktionieren.

Wir merken an, dass im positiv definiten Fall zwar einige Fälle auftreten, in denen $(\mathcal{V}(P))^*$ für eine perfekte quadratische Form P stark entartet ist, diese Kegel aber im Allgemeinen nicht oder nur leicht entartet zu sein scheinen [119, Sektion 10.2]. Es ist zu vermuten, dass sich dies im kopositiven Fall ähnlich verhält. Man beachte dabei, dass wir in Sektion 4.4 sehen werden, dass diese Kegel im kopositiven Fall sehr viel „größer“ werden, als im positiv definiten Fall.

Die double-description Methode

Wie bereits in Sektion 2.2.1 angemerkt, lässt sich ein punktierter polyedrischer Kegel K entweder als Schnitt von Halbräumen, also einem Ungleichungssystem $Ax \geq 0$, oder als die Menge positiver Linearkombination von Elementen $R = (r_1, \dots, r_m)$

$$\{x \in \mathbb{R}^d \mid x = R\alpha, \alpha \geq 0\}.$$

darstellen. Daher erfüllen die Matrizen A und R die folgende Beziehung.

$$x \in K \Leftrightarrow Ax \geq 0 \Leftrightarrow x = R\alpha, \alpha \geq 0. \quad (3.5)$$

Da A und R in einem gewissen Sinne dasselbe Objekt K beschreiben wird (A, R) als „double description pair“, kurz *DD-Paar*, bezeichnet. Ist dabei R so gewählt, dass R eine minimale Anzahl an Spalten hat, also eine minimale erzeugende Menge darstellt, dann sind, wie in der vorherigen Sektion angemerkt, die Spalten von R gerade die Extremalstrahlen von K . Wir wollen also für ein gegebenes A eine Matrix R mit minimaler Spaltenanzahl finden, sodass (A, R) ein DD-Paar sind.

Basierend auf (3.5) entwickelten Motzkin et. al. in [92] die bekannte „Double Description Method“. Es sei dabei erwähnt, dass diese klassische Methode aufgrund ihrer „Natürlichkeit“ über die Jahre immer wieder „neu“ gefunden wurde und verschiedene Namen tragen kann, siehe auch [54].

Die Idee hierbei ist es, für eine initiale Relaxation von K , etwa durch „weglassen“ einiger Zeilen von A , eine minimale Erzeugermatrix R' zu bestimmen, sodass (A', R') ein DD-Paar für den relaxierten Kegel K' ist. Dabei ist A' die Matrix, die aus Weglassen einiger Zeilen von A entsteht. Iterativ erhärtet man dann die Relaxation, um schlussendlich ein DD-Paar (A, R) für den Kegel K zu bekommen, wobei die Spalten von R dann genau die Extremalstrahlen von K sind. Da die Matrix A sowie alle ihre Relaxationen bekannt sind, werden also dabei in jedem Schritt die Extremalstrahlen des erhärteten Kegels aus denen des relaxierten Kegels berechnet.

Wir werden in dieser Sektion der Beschreibung von Malkin in [86, Kapitel 8] folgen. Dafür führen wir zunächst einige Notationen ein. Für ein $I \subseteq \{1, \dots, n\}$ ist A_I die Matrix, die aus A entsteht, wenn man nur Zeilen a_i mit $i \in I$ betrachtet. Damit schreiben wir nun

$$K_I(A) = \{x \in \mathbb{R}^d \mid A_I x = 0, A_I x \geq 0\} \quad \text{und} \quad K_I^i(A) = \{x \in \mathbb{R}^d \mid A_I x = 0, A_{I \setminus \{i\}} x \geq 0\}.$$

Für eine Indexmenge $J \subset I$ schreiben wir dann auch $K_I^J(A) = \bigcap_{j \in J} K_I^j(A)$. Man beachte, dass $K_I^J(A)$ auch als $K_{I \setminus J}(A_J)$ geschrieben werden kann.

Eine geeignete anfängliche Relaxation von $K_I(A)$ ist ein Kegel, für den sich die Extremalstrahlen mit (3.4) „einfach“ bestimmen lassen.

Wenn $K_I(A)$ volldimensional ist, also $I = \{1, \dots, n\}$, bestimmen wir einfach eine Teilmenge $J \subset I$, sodass $|\bar{J}| = d$ und $\text{rk}A_{\bar{J}} = d$. Mit (3.4) folgt dann, dass die Extremalstrahlen von $K_I^J(A) = K_{I \setminus J}(A_J)$ gerade die Spalten von $A_{\bar{J}}^{-1}$ sind [86, Seite 152].

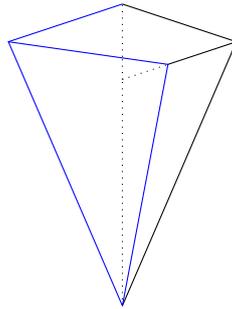
Für den Fall, dass $K_I(A)$ nicht volldimensional ist, muss die vorherige Methode ein wenig angepasst werden. Dafür nehmen wir an, dass $A_{\bar{I}}$ vollen Zeilenrang hat, andernfalls können wir bestimmte Zeilen entfernen ohne $K_I(A)$ zu verändern. Zu \bar{I} fügen wir jetzt so Zeilen J hinzu, dass $\text{rk}A_{\bar{I} \cup J} = d$ und $|\bar{I} \cup J| = d$. Man beachte, dass dies möglich ist, da $K_I(A)$ punktiert ist. Nun können wir wieder $A_{\bar{I} \cup J}$ invertieren, diesmal sind aber nur die Spalten von A^{-1} Extremstrahlen von K_I^J , die nicht zu einem Index in \bar{I} gehören [86, Seite 153].

Man beachte, dass in diesem Fall der Kegel K_I^J $d - \text{rk}A_{\bar{I}}$ Extremalstrahlen hat. Ist $\text{rk}A_{\bar{I}} = d$, so ist $K_I = \{0\}$. Die Bestimmung der initialen Extremalstrahlen hat also in jedem Fall nur polynomiellen Aufwand.

Es bleibt nun noch zu klären, wie wir aus einer Relaxation von $K_I(A)$ die Extremalstrahlen von $K_I(A)$ bestimmen können. Sei dazu $i \in I$, dann ist

$$K_I(A) = K_I^i(A) \cap \{x \in \mathbb{R}^d \mid A_i x \geq 0\}.$$

Daher sind die Extremalstrahlen von $K_I(A)$ gerade die Extremalstrahlen von $K_I^i(A)$, die im offenen Halbraum $\{x \in \mathbb{R}^d \mid A_i x > 0\}$ liegen, zusätzlich zu einigen, die durch den Schnitt jeder 2-Seite von $K_I^i(A)$ mit der Hyperebene $\{x \in \mathbb{R}^d \mid A_i x = 0\}$ erzeugt werden [86, Seite 150].



Um diese neuen Extremalstrahlen zu konstruieren, verwendet man das Konzept der Adjazenz von Extremalstrahlen. Dabei heißen zwei Extremalstrahlen r_1, r_2 des Kegels K *adjazent*, wenn $F = \text{cone}\{r_1, r_2\}$ eine Seite von K ist. Man beachte, dass $\dim F = 2$ ist.

Sind nun r_1, r_2 zwei solcher adjazenter Extremalstrahlen von $K_I^i(A)$, sodass r_1 in $\{x \in \mathbb{R}^d \mid A_i x > 0\}$ und r_2 in $\{x \in \mathbb{R}^d \mid A_i x < 0\}$ liegt. Dann liegt in dem von r_1 und

r_2 erzeugtem Kegel ein Extremalstrahl r von $K_I(A)$. Genauer ist [86, Seite 150]

$$r = \text{cone}((-A_i r_2)r_1 + (A_i r_1)r_2).$$

Man beachte, dass $-A_i r_2 > 0$ ist.

Tatsächlich existiert für jeden Extremalstrahl von $K_I(A)$ so ein Paar von Extremalstrahlen von $K_I^J(A)$ [86, Lemma 8.1.5]. Allerdings ist es effizienter nur die „notwendigen“ Extremalstrahlen so zu konstruieren, sprich nur die Extremalstrahlen, die nicht schon durch die Bedingung $A_i r > 0$ erfasst werden. Damit lässt sich nun der grundlegende Algorithmus angeben, siehe Algorithmus 2.

Input : Kegel $K_I(A)$
Output: Menge der Extremalstrahlen R von $K_I(A)$
 Bestimme wie vorhin beschrieben eine Menge $J \subset I$, so dass $K_I^J(A)$ punktiert und ... ist
 Berechne wie vorhin beschrieben die Extremalstrahlen R von $K_I^J(A)$
while $J \neq \emptyset$ **do**
 Wähle $i \in J$ Setze $R^+ = \{r \in R \mid a_i^T r > 0\}$, $R^- = \{r \in R \mid a_i^T r < 0\}$
 Setze $R = R \setminus R^-$
 forall $r_1 \in R^+$ und $r_2 \in R^-$ **do**
 if r_1 und r_2 sind adjazent **then**
 | Setze $R = R \cup \{(-a_i^T r_2)r_1 + (a_i^T r_1)r_2\}$
 end
 end
 Setze $J = J \setminus \{i\}$
end
 Gib R aus

Algorithmus 2: Grundlegender Algorithmus der double-description Methode, siehe [86, Algorithmus 11]

Um die Adjazenz zweier Extremalstrahlen algorithmisch zu testen, gibt es bisher zwei verschiedene Ansätze, den algebraischen und den kombinatorischen. Für die algebraische Methode benutzt man, dass

$$S = \{x \in \mathbb{R}^d \mid a_i^T x \geq 0 \text{ für } i \in \text{supp}(r_1 + r_2), a_i^T x = 0 \text{ sonst}\}$$

die kleinste Seite von K ist, die F enthält, wobei $\text{supp}(r) = \{i \mid a_i^T(r) > 0\}$. Ist $\dim(S) = 2$, dann ist offenbar $F = S$ und somit r_1 und r_2 adjazent. Anders gesagt testet man ob $\text{rk}(A_{\text{supp}(r_1+r_2)}) = d - 2$ [86, Korollar 8.1.3]. Man beachte, dass man die Rangberechnung frühzeitig abbrechen kann, sobald klar wird, dass $\text{rk}(A_{\text{supp}(r_1+r_2)}) < d - 2$ ist. In Sektion [86, Sektion 8.2.3] werden dabei auch Möglichkeiten besprochen, diesen Ansatz effizienter zu machen, indem man auf bestimmte Weisen die Größe der obigen Matrix reduziert.

Bei der kombinatorischen Methode testet man, ob es einen anderen Extremalstrahl r gibt, sodass $\text{supp}(r) \subseteq \text{supp}(r_1 + r_2) = \text{supp}(r_1) \cup \text{supp}(r_2)$ gilt. Falls es kein solches r

gibt, sind r_1 und r_2 die einzigen Extremalstrahlen in S und daher ist $S = F$. Damit sind r_1 und r_2 adjazent [86, Lemma 8.1.4]. Man beachte, dass nur solche r getestet werden müssen, für die $|\text{supp}(r)| \leq |\text{supp}(r_1 + r_2)|$ ist. Möglichkeiten zur Verbesserung dieser Methode durch Nutzen von Suchbäumen werden in [86, Sektion 8.2.2][118] diskutiert.

Da die algebraische Methode auf der Rangberechnung einer Matrix fußt, ist sie zu bevorzugen, wenn die Eingabe relativ „klein“ ist, andernfalls sollte man die kombinatorische Methode verwenden [86, Seite 153]. Man beachte, dass die Effizienz beider Methoden auch sehr stark von der Implementation abhängt [86, Seite 153].

In [86, Sektion 8.2.1] wurden weitere Methoden vorgestellt, mit denen man schnell bestimmte nicht-adjazente Paare von Extremalstrahlen eliminieren kann sowie für bestimmte Paare schnell die Adjazenz erhält. Diese sind wohl für praktische Implementationen unabdingbar, weshalb wir sie hier kurz angeben wollen.

Zunächst sind zwei Extremalstrahlen r_1 und r_2 nicht adjazent, wenn $|\text{supp}(r_1 + r_2)| > n - d + 2$ ist [86, Lemma 8.2.1]. Die Idee hierbei ist, dass dann der Vektor $r_1 + r_2$ zu wenige der Ungleichungen mit Gleichheit erfüllt, um in einer 2-Seite von K zu liegen, woraus folgt, dass r_1 und r_2 keine 2-Seite aufspannen.

Weiter sind, mit ähnlichen Ideen, die Extremstrahlen r_1 und r_2 adjazent, wenn $|\text{supp}(r_1) \setminus \text{supp}(r_2)| = 1$ oder $|\text{supp}(r_2) \setminus \text{supp}(r_1)| = 1$ ist [86, Lemma 8.2.2]. Erfüllt einer der Extremalstrahlen genau $d - 1$ Gleichungen mit Gleichheit, so ist dies eine notwendige und hinreichende Bedingung [86, Lemma 8.2.3].

Man beachte, dass sich diese Bedingungen schnell überprüfen lassen, sodass damit tatsächlich schnell die Adjazenz von bestimmten Extremalstrahlen getestet werden kann.

Ein Faktor, der sich ebenfalls auf die Laufzeit auswirken kann, ist die Reihenfolge der ausgewählten Indizes $i \in J$. In [54] werden drei verschiedene Möglichkeiten präsentiert. Einerseits ist eine lexikographische Wahl von i möglich, dies wird auch statische Wahl genannt. Andererseits kann man i so wählen, dass R^- beziehungsweise R^+ größtmöglich werden, „max-cut-off“ beziehungsweise „min-cut-off“ genannt. Dies ist eine dynamische Wahl. Den Autoren von [54] zufolge scheint die lexikographische Wahl besser zu funktionieren, [86, Seite 153] zufolge hängt dies sehr stark von der jeweiligen Instanz ab.

In [86, Sektion 8.1] wird weiter die sogenannte „max-intersection“-Wahl präsentiert, wobei i so gewählt wird, dass die Menge $\{r \mid A_i r = 0\}$ möglichst groß wird.

Die Komplexität und der zeitliche Aufwand der double description method ist noch nicht gut verstanden. Man beachte, dass die einzelnen Schritte von Algorithmus 2 polynomiellen Aufwand haben, daraus allerdings nicht folgt, dass Algorithmus 2 polynomiell ist, da die Zwischenkegel exponentiell größer als der ursprüngliche Kegel sein können.

Welche der Zwischenkegel konstruiert werden, ist nur von der Wahl von $i \in J$ abhängig. Tatsächlich konnte in [5] gezeigt werden, dass es für Wahlen von i die die Informationen der Eingabe nicht verwendet immer Klassen von Kegeln gibt, die im schlimmsten Fall exponentiell große Zwischenkegel konstruiert. Es ist noch ein offenes Problem, ob es eine Regel zur Wahl von i gibt, mit der Algorithmus 2 polynomiell in Eingabe- und Ausgabegröße ist [86, Seite 146].

Der reverse-search-Algorithmus

Wir beschreiben hier den reverse-search-Algorithmus nach [3] für den Fall eines allgemeinen Polyeders $P = \{y \in \mathbb{R}^d \mid Ay+b \geq 0\}$. Man beachte, dass punktierte polyedrische Kegel Polyeder mit genau einer Ecke, dem Ursprung, sind.

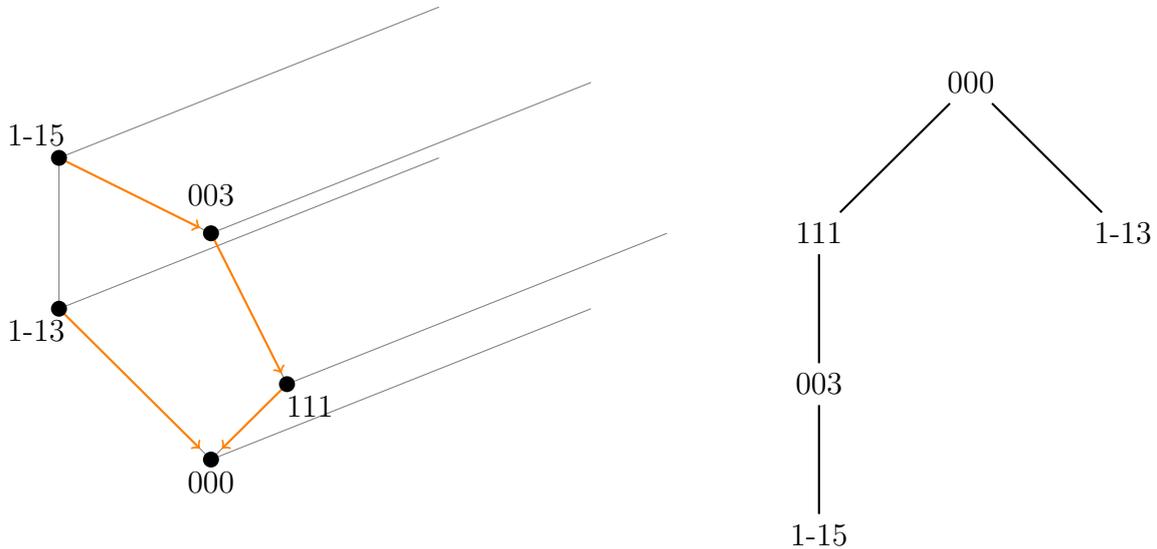


Abbildung 3.3: Verlauf der Simplexmethode bei verschiedenen Startpunkten bezüglich der Zielfunktion $-(x_1 + x_2 + x_3)$

Wir wollen die Idee des reverse-search-Algorithmus anhand obiger Abbildung beschreiben. Die Abbildung beschreibt den Verlauf der klassischen Simplexmethode zum Lösen linearer Optimierungsprobleme (siehe auch [110, Kapitel 11]). Dabei geht die Methode, von einer beliebigen Startecke aus, dem Baum, auf der rechten Seite, „nach oben“ in Richtung Wurzel durch, um schlussendlich bei einer optimalen Ecke, der Wurzel, anzukommen.

Dieses Verhalten macht sich der reverse-search Algorithmus zunutze, indem er diesen Prozess umkehrt. Man startet also bei einer optimalen Lösung, der Wurzel, und geht den Baum „nach unten“ durch und erhält so nach und nach alle Ecken (und etwaigen Extremalstrahlen) des Polyeders.

Im Folgenden gehen wir genauer auf die Funktionsweise des reverse-search Algorithmus ein.

Zunächst können wir den Polyeder P durch

$$P = \left\{ y \in \mathbb{R}^d \mid y_i = x_i, i = 1, \dots, d, x_{d+i} = b_i + \sum_{j=1}^d a_{ij}x_j, x_{d+i} \geq 0, i = 1, \dots, n, x \in \mathbb{R}^{d+n} \right\}$$

darstellen [3, Sektion 3]. Dabei können wir x_{d+i} , $i = 1, \dots, n$ als Schlupfvariablen auffassen.

Nach etwaiger Umsortierung der Zeilen von A (man beachte, dass dies P nicht verändert), fassen wir die Variablen x_{n+i} , $i = 1, \dots, d$ mit der Terminologie der Simplex-

methode als Nichtbasis-Variablen auf und stellen die übrigen (Basis-)Variablen durch diese dar. Damit erhält man [3, Gleichung (3.1)]

$$x_i = b'_i + \sum_{j=1}^d a'_{ij} x_{n+j}, \quad i = 1, \dots, n,$$

für geeignete b'_i und a'_{ij} . Diese Gleichungen geben eine äquivalente Darstellung für P an. Zu der daraus entstandenen Matrix A' fügen wir die „0-te“ Zeile $(-1, \dots, -1)$ hinzu. Dies sichert ab, dass wir die Wurzel unseres Suchbaumes in jedem Fall kennen, wie wir später sehen werden.

Bezeichnet nun $\bar{A} = (I \ -A')$, $\bar{b} = (0 \ b')^T$ und A_I die Matrix die aus A durch Betrachtung der Spalten mit Index in I betrachtet. Sei B eine geordnete Menge von $n+1$ Indizes aus $\{d+n+1\}$ so, dass \bar{A}_B nicht singular ist und bezeichne N die Menge der restlichen Indizes. Dann lassen sich die obigen Gleichungen als

$$x_B + \bar{A}_B^{-1} \bar{A}_N x_N = \bar{A}_B^{-1} \bar{b} \quad (3.6)$$

schreiben. (3.6) wird *Verzeichnis* genannt, die Mengen B Basis und N Cobasis. Wie wir gleich sehen werden, besteht der Algorithmus aus Manipulationen von Verzeichnissen. Solche Verzeichnisse können auch äquivalent durch die Matrix

$$\begin{pmatrix} \bar{A}_B^{-1} & \bar{A}_N \end{pmatrix} \quad (3.7)$$

dargestellt werden.

Eine Basis B und eine Lösung x des Verzeichnisses (3.6) bezeichnen wir als zulässig, falls

$$\{0, \dots, d\} \subset B \text{ und } x_i \geq 0, \quad i \in B \cap \{d+1, \dots, d+n\} \text{ ist.} \quad (3.8)$$

Sind beide Bedingung erfüllt, so nennen wir (3.6) zulässig. Ist x eine zulässige Lösung, so nennen wir den Vektor, den wir erhalten, wenn wir $x_N = 0$ setzen, zulässige Basislösung.

Diese zulässigen Verzeichnisse kann man nun benutzen, um die Ecken und Extremalstrahlen des Polyeders P darzustellen [3, Proposition 3.1, 3.2]. Sei dafür ein zulässiges Verzeichnis mit Basis B und Cobasis N gegeben. Indem wir $y_i = x_i$, $i = 1, \dots, d$ für die zulässige Basislösung x setzen, erhalten wir eine Ecke von P , da durch N gerade d Ungleichungen definiert werden, welche y mit Gleichheit erfüllt. Mit ähnlichen Ideen erhält man, dass wir für jeden Extremalstrahl $y + r$ von P eine zulässige Basis B und einen Cobasisindex i haben, sodass r durch $\bar{A}_B^{-1} A_i$ gegeben wird.

Das Problem der Bestimmung aller Ecken und Extremalstrahlen ist also äquivalent zum Finden aller zulässigen Basen. Dabei kann es vorkommen, dass es für eine Ecke oder einen Extremalstrahl mehr als nur eine solche zulässige Basis gibt. Genauer ist dies gerade dann der Fall, wenn der Polyeder entartet ist [86, Seite 146]. Daher kann es vorkommen, dass mit dieser Methode eine Ecke oder Extremalstrahl mehrfach gefunden und ausgegeben wird.

Um dieses Problem zu händeln, wurden in [3, Sektion 4,5] bestimmte Klassen von Basen, sogenannte „lex-positive“ und „lex-minimale“ Basen, eingeführt. Dabei heißt

eine zulässige Basis *lex-positiv*, falls die ersten Nicht-Null-Elemente der Spalten mit Index $d + 1, \dots, n$ der Matrix $\begin{pmatrix} \bar{A}_B^{-1} \bar{b} & \bar{A}_B^{-1} \end{pmatrix}$ positiv sind.

Weiter heißt für eine zulässige Basislösung die zugehörige Basis B *lex-minimal*, wenn sie die lexikographisch kleinste Basis von x ist, es also keine Basis B' von x gibt, deren kleinster Index kleiner als der von B ist. Tatsächlich sind lex-minimale Basen, die Ecken und Extremalstrahlen darstellen, eindeutig und lex-positiv [3, Theorem 5.5]. Es genügt also, solche lex-positiven Basen zu betrachten und zu finden.

Wir wollen nun beschreiben, wie wir solche Basen und Verzeichnisse finden. Zunächst definieren wir die Wurzel des Suchbaums (siehe Abbildung 3.3) durch die Basis $B^* = \{0, \dots, n\}$ und Cobasis $N^* = \{m + 1, \dots, m + d\}$. Das daraus entstehende Verzeichnis hat die Form $\begin{pmatrix} I & \bar{A}_N \end{pmatrix}$. Diese Basis ist offenbar lex-positiv. Von dieser Basis aus gehen wir dann die Äste des Baumes durch, um alle lex-minimalen Basen zu finden. Wurde ein Ast komplett abgearbeitet, kehrt man dann wieder zur Wurzel zurück und durchsucht den nächsten Ast.

Um auf diese Weise von einer Basis zur nächsten zu kommen, machen wir bestimmte „Pivotschritte“. Ein *Pivotschritt* besteht daraus, einen Basisindex i durch einen Cobasisindex j zu ersetzen. Dieser Schritt wirkt sich auf das Verzeichnis (3.7) wie folgt aus: Sei zunächst $a = \bar{A}_B^{-1} A_j$ und sei der Basisindex i an der k -ten Stelle in B , dann teilen wir die k -te Zeile des Verzeichnisses (3.7) durch a_k und ziehen a_i * Neue Zeile k von den restlichen Zeilen ab. Man beachte, dass diese Operation gerade der Pivotschritt-Operation der Simplexmethode entspricht [110, Sektion 11.2].

Bezeichnet B' die durch diesen Pivotschritt erhaltene Basis, so entsteht aus der obigen Operation gerade das Verzeichnis für die Basis B' [3, Proposition 3.3].

Damit für einen Cobasisindex j ein geeigneter Basisindex i für einen Pivotschritt gewählt wird, wird in [3] die bekannte „lexikographische Regel“ der Simplexmethode verwendet, siehe auch [110, Sektion 11.3]. Dabei wählen wir für ein $j \in N$ i , so dass er der kleinste Index zwischen $d + 1$ und n mit $a_i > 0$ ist. Dabei ist $a = \bar{A}_B^{-1} A_j$. Gibt es keinen solchen Index, so setzt man $i = 0$.

In der Simplexmethode hat diese Regel den Zweck, Kreise zu verhindern [110, Sektion 11.3]. Für den reverse-search Algorithmus ergeben sich aus dieser Regel zwei Vorteile. Ist zunächst B eine lex-positive Basis, dann ist die durch diesen Pivotschritt entstandene Basis B' ebenfalls lex-positiv [3, Proposition 4.1 (a)]. Da wir weiter, wie oben beschrieben, nachdem wir einen Ast vollständig abgehandelt haben zur Wurzel zurückkehren wollen, ergibt sich mit dieser Pivotregel, dass für den Cobasisindex $i \in N'$ der Basisindex $j \in B'$ als „Pivotelement“ gewählt wird [3, Proposition 4.1 (b)]. Dies ermöglicht es also, den Baum in die andere „Richtung“ zu durchlaufen.

Damit können wir nun eine grundlegende Version des reverse-search Algorithmus angeben, siehe Algorithmus 3.

In [3, Sektion 5,6] werden effiziente Methoden zum Überprüfen der ersten **if**-Bedingung in Algorithmus 3 sowie der Lex-Minimalität einer Basis für eine Ecke oder einen Extremstrahl entwickelt [3, Proposition 5.1,5.4,6.1]. Auf diese wollen wir hier nicht weiter eingehen, stattdessen verweisen wir auf [3, Sektion 5,6], sowie [3, Sektion 7] und [4] für

Input : Polyeder P , welcher durch die Matrix A beschrieben wird
Output: Alle Ecken und Extremalstrahlen von P
 Setze $B = \{0, \dots, n\}$, $N = \bar{B}$, $k = 1$
repeat
 while $k \leq d$ **do**
 Setze $j = N_k$
 if Es gibt einen Index i in B , sodass B durch einen Pivotschritt von $B \setminus \{i\} \cup \{j\}$ erreicht wird **then**
 Führe Pivotschritt mit i und j durch
 if B ist lex-minimal **then**
 Gib entsprechende Ecke aus
 Setze $k = 1$
 end
 else
 Setze $k = k + 1$
 end
 end
 Finde Pivotelemente $i = B_r, j = N_k$ und führe Pivotschritt durch
 Setze $k = k + 1$
until $k > d$ und $B = \{0, \dots, n\}$;

Algorithmus 3: Grundlegender reverse-search Algorithmus, siehe [3, Abbildung 6.1]

genauere Implementationsdetails.

Im Gegensatz zur double-description Methode ist der Aufwand des reverse-search Algorithmus bekannt, so berechnet er alle Extremalstrahlen und Ecken eines Polyeders mit einem zeitlichen Aufwand von $\mathcal{O}(nd^2)$ pro Basis und einem Speicheraufwand von $\mathcal{O}(nd)$ [3, Theorem 6.2]. Ein weiterer Vorteil des reverse-search Algorithmus ist, dass er sich gut parallelisieren lässt [28], da man jeden Ast des Suchbaums, wie in Abbildung 3.3, unabhängig voneinander durchgehen kann.

Mögliche Pivotregeln

Wir haben nun gesehen, wie wir in der **while**-Schleife von Algorithmus 1 die Extremalstrahlen der Kegel $(\mathcal{V}(P))^*$ bestimmen können. Nun stellt sich die Frage, wie wir abhängig von der Matrix Q einen möglichst „guten“ Extremalstrahl R mit $\langle R, Q \rangle < 0$ auswählen. Man beachte, dass es immer einen solchen Extremalstrahl gibt [110, Theorem 7.1].

Dieses Auswahlkriterium bezeichnen wir auch als *Pivotregel*. In [47, Sektion 3.1] haben die Autoren durch Experimente zwei scheinbar gute Heuristiken für eine Pivotregel gefunden. Sowohl die Wahl von R , sodass $\frac{R'}{\|R'\|}$ minimal unter den möglichen Extremalstrahlen, als auch eine zufällige Wahl von R scheinen in Tests gut zu funktionieren.

Allerdings ist die Frage nach einer möglichst „guten“ Pivotregel ein noch offenes Problem.

3.2.2 Bestimmung einer benachbarten \mathcal{COP} -perfekten Matrix

Die Hauptkomplexität eines einzelnen Schrittes von Algorithmus 1 liegt in der Bestimmung einer benachbarten \mathcal{COP} -perfekten Matrix. Algorithmisch können wir dies ähnlich wie im Falle der perfekten quadratischen Formen erreichen [111, Sektion 3.1.5, Algorithmus 2]. Dabei nennen wir zwei \mathcal{COP} -perfekte Matrizen P und N *benachbart*, falls die Strecke $[P, N]$ eine Kante von \mathcal{R} ist.

Dazu muss der im positiv definiten Fall verwendete Algorithmus etwas verändert werden, siehe Algorithmus 4. Im Folgenden betrachten wir daher die Funktionsweise dieses, für Algorithmus 1 zentralen, Algorithmus. Dabei folgen wir der Beschreibung in [47].

Input : \mathcal{COP} -perfekte Matrix P , Extremalstrahl $R \notin \mathcal{COP}^d$ von $(\mathcal{V}(P))^*$
Output: \mathcal{COP} -perfekte Matrix $N = P + \lambda R$ mit $\lambda > 0$, $\min_{\mathcal{COP}} N = 1$ und $\min_{\mathcal{COP}} N \not\subseteq \min_{\mathcal{COP}} P$

Setze $(l, u) = (0, 1)$

while $P + uR \notin \text{int } \mathcal{COP}^d$ oder $\min_{\mathcal{COP}}(P + uR) = 1$ **do**

if $P + uR \notin \text{int } \mathcal{COP}^d$ **then**

 | Setze $u = (l + u)/2$

else

 | Setze $(l, u) = (u, 2u)$

end

end

Bestimme $S = \{v \in \mathbb{Z}_{\geq 0}^d \setminus \{0\} \mid (P + uR)[v] < 1\}$

Setze $\lambda = \min \left\{ \frac{1 - P[v]}{R[v]} \mid v \in S \right\}$

Setze $N = P + \lambda R$

Setze

$\min_{\mathcal{COP}} N = (\min_{\mathcal{COP}} P \setminus \{v \in \min_{\mathcal{COP}} P \mid R[v] \neq 0\}) \cup \{v \in S \mid \frac{1 - P[v]}{R[v]} = \lambda\}$

Gib N und $\min_{\mathcal{COP}} N$ aus

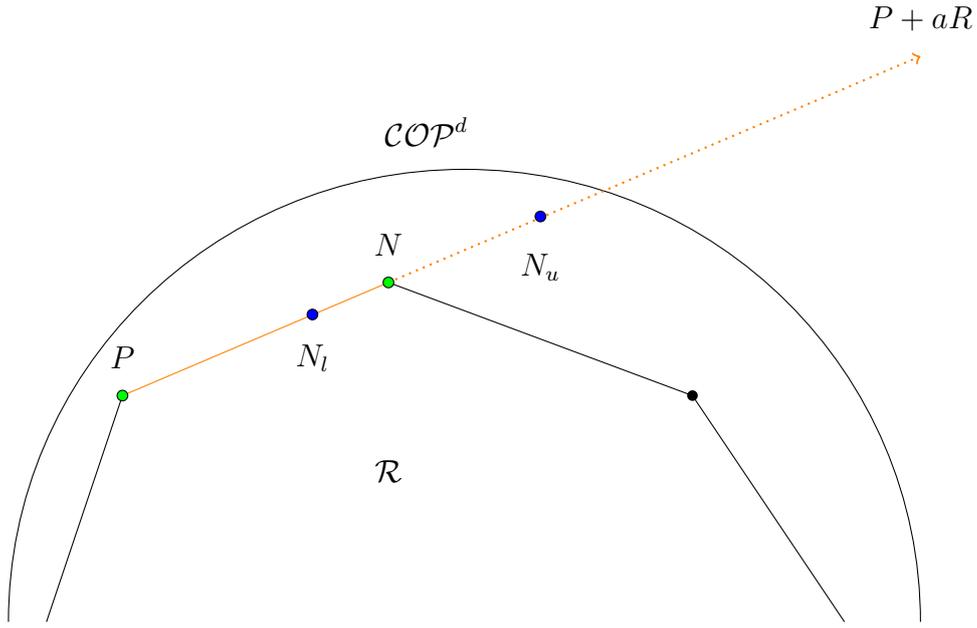
Algorithmus 4: Algorithmus zum Bestimmen einer benachbarten \mathcal{COP} -perfekten Matrix, siehe [47, Algorithmus 2]

Zunächst bildet der Schnitt der „Halbgeraden“ $\{P + aR \mid a \geq 0\}$ mit \mathcal{R} für einen nicht-kopositiven Extremalstrahl R von $(\mathcal{V}(P))^*$ eine Kante von \mathcal{R} . Dies folgt sofort mit (3.4), da diese Halbgerade im Schnitt von $\binom{d+1}{2}$ Hyperebenen liegt.

Daher liegt auf der Halbgeraden $\{P + aR \mid a \geq 0\}$ eine mit P benachbarte \mathcal{COP} -perfekte Matrix N .

Algorithmus 4 konstruiert nun sukzessiv Matrizen $N_l = P + lR$, die auf der Kante $[P, N]$ liegt und $N_u = P + uR$, die außerhalb von \mathcal{R} aber noch in $\text{int } \mathcal{COP}^d$ liegt. l und u geben dann eine untere und obere Schranke für den Wert λ an, für den $N = P + \lambda R$ \mathcal{COP} -perfekt ist. Man beachte, dass $\min_{\mathcal{COP}} N_l = \min_{\mathcal{COP}} P = 1 > \min_{\mathcal{COP}} N_u$ ist.

Die Schwierigkeit von Algorithmus 4 besteht nun aus der Überprüfung der Kopositivität von N_u und aus der Berechnung des kopositiven Minimums einer Matrix $Q \in \text{int } \mathcal{COP}^d$,



sowie der Menge S . Algorithmische Kopositivitätstest besprechen wir in der nächsten Sektion. Auf die Berechnung des kopositiven Minimums und der Menge S gehen wir in Kapitel 5 ein. Wir zeigen im Folgenden die Korrektheit von Algorithmus 4 wie in [47, Sektion 3.2] und identifizieren einige Schwierigkeiten und geben mögliche Verbesserungen an.

Die Menge S im Algorithmus 4 besteht anschaulich aus den Vektoren, die „verhindern“, dass $\min_{\mathcal{COP}} N_u = 1$ ist. Man beachte, dass S endlich ist. Setzen wir für ein $v \in S$, $u_v = \frac{1-P[v]}{R[v]}$, so besteht die neue Menge gerade aus den Vektoren w in S für die $\frac{1-P[w]}{R[w]} < \frac{1-P[v]}{R[v]}$ ist, wie wir gleich sehen werden. Daher ist für $P + \lambda R$ mit $\lambda = \min_{w \in S} \frac{1-P[w]}{R[w]}$ die zugehörige Menge $S = \emptyset$.

Es bleibt also zunächst zu sehen, dass für dieses N $\min_{\mathcal{COP}} N = 1$ und $\text{Min}_{\mathcal{COP}} N \not\subseteq \text{Min}_{\mathcal{COP}} P$ ist. Für den zweiten Teil betrachten wir ein $v \in S$, so dass $\frac{1-P[v]}{R[v]}$ minimal ist. Dann ist nach Konstruktion $\lambda = \frac{1-P[v]}{R[v]}$ und es ist

$$N[v] = (P + \lambda R)[v] = P[v] + \frac{1 - P[v]}{R[v]} = P[v] + 1 - P[v] = 1.$$

Weiterhin gilt für alle $v \in S$, dass $R[v] < 0$, da $\min_{\mathcal{COP}} P = 1$ und $u > 0$. Wäre nun $P[v] = 1$, also $v \in \text{Min}_{\mathcal{COP}} P$, so ist nach Definition $R[v] \geq 0$ und damit $v \notin S$.

Tatsächlich gilt für alle $v \in S$ für die $\frac{1-P[v]}{R[v]}$ minimal ist, dass $N[v] = 1$ und $v \notin \text{Min}_{\mathcal{COP}} P$ ist. Ist also $\min_{\mathcal{COP}} N = 1$, so gilt $\text{Min}_{\mathcal{COP}} N \not\subseteq \text{Min}_{\mathcal{COP}} P$. Dadurch erhalten wir zusätzlich $\text{Min}_{\mathcal{COP}} N$, denn damit ist

$$\text{Min}_{\mathcal{COP}} N = \text{Min}_{\mathcal{COP}} P \setminus \{v \in \text{Min}_{\mathcal{COP}} P \mid R[v] \neq 0\} \cup \left\{ v \in S \mid \frac{1 - P[v]}{R[v]} = \lambda \right\}. \quad (3.9)$$

Für das kopositive Minimum von N betrachten wir die Fälle $v \notin S$ und $v \in S$. Für $v \notin S$ gilt entweder $R[v] \geq 0$, in diesem Fall haben wir

$$(P + \lambda R)[v] = P[v] + \lambda R[v] \geq 1,$$

da $\lambda > 0$ ist, oder $R[v] < 0$. In diesem Fall

$$(P + \lambda R)[v] = P[v] + \lambda R[v] \geq P[v] + uR[v] \geq 1.$$

Dies gilt, da $\lambda \leq u$ ist.

Für $v \in S$ wissen wir, dass $R[v] < 0$ ist. Weiter gilt $R[v] \geq \min_{w \in S} R[w]$ und $1 - P[w] < 0$ für alle $w \in S$. Damit erhalten wir, dass $\min_{w \in S} \frac{1-P[w]}{R[w]} \geq \frac{1-P[v]}{R[v]}$ ist und somit

$$(P + \lambda R)[v] = P[v] + \min_{w \in S} \frac{1 - P[w]}{R[w]} R[v] \geq P[v] + \frac{1 - P[v]}{R[v]} R[v] = 1.$$

Daher ist also $\min_{\mathcal{COP}} N = 1$.

Die Menge S in Algorithmus 4 kann unter Umständen sehr groß werden. So hat sich im Fall der positiv definiten Matrizen gezeigt, dass die Menge S um den Faktor $\frac{1}{\lambda(N_u)} \mathcal{O}(d)$ wächst [119, Sektion 5]. Daher ist S , falls $\lambda(N_u) < 1$ sehr klein ist, sehr groß. Auch hier ist anzunehmen, dass es sich im kopositiven Fall recht ähnlich verhält. Daher ist die vollständige Berechnung von S in gewissen Fällen sehr aufwendig beziehungsweise schlicht nicht möglich.

Eine Möglichkeit dieses Problem zu umgehen ist, iterativ immer nur einen Vektor v in S zu ermitteln und u sukzessiv mittels $u = \frac{1-P[v]}{R[v]}$ zu reduzieren, solange bis wir $\min_{\mathcal{COP}} N_u = 1$ erhalten. Man beachte, dass sich dadurch S verändert. Dann ist nach den obigen Beobachtungen $u = \lambda$ und $N = N_u$ \mathcal{COP} -perfekt.

Die Methode ermöglicht also, dass wir die ursprüngliche Menge S nicht vollständig berechnen müssen. Der Nachteil dieser Methode ist, dass wir (3.9) nicht anwenden können und somit $\text{Min}_{\mathcal{COP}} N$ explizit berechnen müssen. Dies lässt sich aber möglicherweise recht effizient mit einer ähnlichen Strategie wie in [119, Algorithmus 3] umgehen.

```

while  $\min_{\mathcal{COP}} N_u \leq \frac{1}{2}$  do
  | Bestimme ein  $v$  mit  $N_u[v] < 1$ 
  | Setze  $u = \frac{1-P[v]}{R[v]}$ 
end
Bestimme  $S$  wie in Algorithmus 4

```

Die Idee dabei ist, u so lange zu reduzieren, bis $\min_{\mathcal{COP}} N_u$ „groß“ genug ist, oder äquivalent S „klein“ genug ist. Dann kann man S mit der direkten Methode berechnen. Man beachte, dass in der **while**-Schleife keine vollständige Berechnung des kopositiven Minimums stattfinden muss, solange $\min_{\mathcal{COP}} N_u \leq \frac{1}{2}$ ist, da dann lediglich ein Vektor v mit $N_u[v] \leq \frac{1}{2}$ ausreicht. Eine vollständige Berechnung von S ist also nur im letzten Schritt der **while**-Schleife notwendig.

Damit dieser Ansatz allerdings wirklich gut funktionieren kann, bräuchte es, wie im positiv definiten Fall [119, Algorithmus 3], einen Algorithmus, der „schnell“ einen minimalen Vektor von N_u berechnen kann. Uns ist kein solcher Algorithmus bekannt, sodass die Entwicklung eines solchen durchaus interessant wäre.

Eine weitere einfache Verbesserung geben uns die in Kapitel 5 vorgestellten Methoden zur Berechnung von des kositiven Minimums, welche zusätzlich $\text{Min}_{\mathcal{COP}} Q$ bestimmen. Wir wissen, dass für alle $N, P \neq Q \in [P, N]$ gilt, dass $\text{Min}_{\mathcal{COP}} Q \subset \text{Min}_{\mathcal{COP}} P, \text{Min}_{\mathcal{COP}} N$. Wenn also $\text{min}_{\mathcal{COP}} N_u = 1$ und $\text{Min}_{\mathcal{COP}} N_u \not\subset \text{Min}_{\mathcal{COP}} P$ ist, dann ist $\lambda = u$ und $N = N_u$. In unserer Erfahrung tritt der Fall $u = \lambda$ in einem Schritt der **while**-Schleife durchaus häufiger ein, bei geeigneter Skalierung des Extremalstrahles R . Ein zusätzlicher Test von $\text{Min}_{\mathcal{COP}} N_u \not\subset \text{Min}_{\mathcal{COP}} P$ kann daher, ohne zusätzlichen Aufwand, Algorithmus 4 vorzeitig beenden und die aufwendige Berechnung der Menge S , sowie weitere „unnötige“ Schleifendurchläufe verhindern.

Für eine weitere Verbesserung von Algorithmus 4 könnte die Beantwortung der folgenden Frage hilfreich sein.

Frage. Ist es möglich eine kopositive Version von [119, Lemma 9] zu konstruieren?

[119, Lemma 9] gibt eine Konstruktion von u für den positiv definiten Fall an, sodass $P + uR$ ein arithmetisches Minimum von < 1 hat, aber noch in $\mathcal{S}_{>0}^d$ liegt. Eine kopositive Version von [119, Lemma 9], also eine Konstruktion von u , sodass $\text{min}_{\mathcal{COP}} P + uR < 1$ und $P + uR \in \mathcal{COP}^d$, würde die **while**-Schleife in Algorithmus 4 unnötig machen und somit wiederholte Kositivitätstests und Berechnungen des kositiven Minimums verfallen lassen.

3.2.3 Kopositivitätstests

Wie wir in der vorherigen Sektion gesehen haben ist es für Algorithmus 4 und damit für Algorithmus 1 notwendig, algorithmisch eine Matrix auf Kopositivität beziehungsweise strikte Kopositivität zu testen. Auch der in Kapitel 4 behandelte Algorithmus 6 benötigt diese Tests.

Daher beschreiben wir im Folgenden wie sich dies realisieren lässt. Zunächst gehen wir dafür auf die Komplexität dieses Problems ein und beschreiben dann einfache Kopositivitätskriterien für kleine Dimensionen. Danach werden wir grundlegende Vorbereitungstechniken für Kopositivitätstests einführen und beschreiben dann einen klassischen Kopositivitätstest sowie einen erst kürzlich aufgetretenen.

Komplexität

Murty und Kabadi zeigten in [94, Theorem 3], dass das Entscheidungsproblem „Gegeben eine Matrix Q , ist Q nicht kopositiv?“ \mathcal{NP} -vollständig ist. Dies folgt wesentlich daraus, dass man \mathcal{NP} -vollständige „SUBSETSUM“-Problem [73, Main Theorem] auf das äquivalente Problem „Gibt es ein $x \in \Delta^S$, sodass $Q[x] < 0$ ist?“ reduzieren kann [94, Theorem 2]. Daher ist das komplementäre Entscheidungsproblem „Gegeben eine Matrix Q , ist Q kopositiv“ $\text{co-}\mathcal{NP}$ -vollständig. Dieses Problem wird auch das „Strong Membership Problem“ des Kegels \mathcal{COP}^d [43, Definition 3.2] genannt. Man beachte, dass dieses äquivalente Problem auch durch ein Standard-Quadratisches Problem (2.11) dargestellt werden kann.

Dickinson und Gijben erweiterten dieses Resultat und zeigten, dass das obige Problem \mathcal{NP} -vollständig ist [43, Theorem 4.2]. Insbesondere liegt das Strong-Membership-Problem also in $\mathcal{NP} \cap \text{co-}\mathcal{NP}$.

Damit sehen wir nun auch, dass die Komplexität der umformulierten schwierigen Probleme in Sektion 2.3.2 tatsächlich in Bedingung $X \in \mathcal{COP}^d$ „geschoben“ wird [48, Seite 5].

Da die Kegel \mathcal{CP}^d und \mathcal{COP}^d dual zueinander sind, siehe Sektion 2.2, ist anzunehmen, dass das Strong Membership Problem des Kegels \mathcal{CP}^d eine ähnliche Komplexität hat. Dies werden wir in Sektion 4.1 betrachten.

Kopositivität in kleinen Dimensionen und für spezielle Matrizen

Auch wenn es im Allgemeinen sehr schwierig ist zu entscheiden ob eine Matrix kopositiv ist, gibt es mehrere Spezialfälle, für die es sehr einfach ist.

So ist eine Matrix $Q \in \mathcal{S}^2$ kopositiv, wenn [65, Proposition 2.1]

$$q_{11}, q_{22} \geq 0 \text{ und } q_{12} + \sqrt{q_{11}q_{22}} \geq 0. \quad (3.10)$$

Man beachte, dass diese Bedingung äquivalent dazu ist, dass Q auf dem Standardsimplex Δ^S nichtnegativ ist. Dies ist eine ausreichende Bedingung für Kopositivität, welche uns in Sektion 3.2.3 und 5.1.1 begegnen wird.

Für Dimension 3 und 4 lassen sich ähnliche, wenn auch länglichere Aussagen treffen, so müssen für eine Matrix $Q \in \mathcal{S}^3$ sieben Ungleichungen, die alle recht ähnlich zu denen

im zweidimensionalen Fall sind, gelten [65, Proposition 2.2]. Li und Feng geben in [83] weitere Kriterien für den dreidimensionalen Fall an sowie Kriterien für den vierdimensionalen Fall [83, Sektion 4], auf die wir hier aber nicht weiter eingehen wollen.

Weitere effiziente Kopositivitätstest gibt es für Matrizen mit bestimmten Strukturen. Ein Werkzeug, welches häufig für theoretische Resultate über positiv semidefinite Matrizen verwendet wird, ist das sogenannte Schur-Komplement. Dabei heißt für eine Matrix $Q = \begin{pmatrix} a & b^T \\ b & C \end{pmatrix}$ die Matrix $aC - bb^T$ das *Schur-Komplement* von Q . Es lässt sich einfach zeigen, dass Q genau dann positiv semidefinit ist, wenn $a \geq 0$ ist und C und $aC - bb^T$ positiv semidefinit sind [14].

Für kopositive Matrizen lässt sich eine recht ähnliche Aussage treffen. So ist Q genau dann kopositiv, wenn $a \geq 0$ ist, C kopositiv ist und für alle $x \in \mathbb{R}_{\geq 0}^{d-1}$ mit $b^T x \leq 0$ gilt, dass $(aC - bb^T)[x] \geq 0$ ist [83, Theorem 2]. Eine wichtige Beobachtung hierbei ist, dass sich die letzte Bedingung zu

$$\begin{pmatrix} a & b^T x \\ b^T x & C[x] \end{pmatrix} \in \mathcal{COP}^2 \text{ für alle } x \in \mathbb{R}_{\geq 0}^{d-1}$$

umformen lässt und man (3.10) darauf anwenden kann.

Daraus ergeben sich zwei Folgerungen. Ist b ein nicht-positiver Vektor, so gilt $b^T x \leq 0$ für alle nichtnegativen Vektoren x . Daher ist die dritte Bedingung in diesem Fall, dass $aC - bb^T$ kopositiv ist.

Die andere Folgerung benötigen wir in Sektion 4.3, weshalb wir sie hier in Form eines Lemmas aufführen werden.

Lemma 3.2. *Eine symmetrische Matrix Q , welche nur positive Diagonal- und nur nichtpositive Außerdiagonaleinträge besitzt, ist genau dann kopositiv, wenn sie positiv semidefinit ist*

Die für uns wichtige Eigenschaft die sich daraus ergibt ist, dass $\mathcal{COP}^2 = \mathcal{S}_{\geq 0}^2 \cup \mathcal{N}^2$ ist, was eine Verschärfung von (2.2.3) darstellt.

Das Schur-Komplement-Argument erlaubt es also in der Theorie spezielle Matrizen auf Kopositivität zu testen. Praktisch kann es beispielsweise für lineare Kopositivitätstests tridiagonaler Matrizen benutzt werden, wie Bomze in [17] zeigte. Diese Idee lässt sich auf Blocktridiagonalmatrizen [17] und ebenso auf allgemeine azyklische Matrizen [68] verallgemeinern. Dabei ist eine symmetrische Matrix azyklisch, wenn ihr Matrixgraph azyklisch ist. Man beachte, dass daher jede Tridiagonalmatrix azyklisch ist.

Für eine Verbesserung von Algorithmus 4 wäre es möglicherweise interessant zu überlegen, ob man die Struktur der Matrizen in Algorithmus 4 für einen Kopositivitätstest ausnutzen kann. Man beachte, dass P immer strikt kopositiv und \mathcal{COP} -perfekt ist und R eine indefinite nicht kopositive Matrix ist.

Vorbereitung

Wir haben nun gesehen, dass sich in speziellen Fällen Kopositivität effizient testen. Wie wir im Folgenden sehen werden, lässt sich ebenfalls für eine Vielzahl von Fällen die Kopositivität effizient ausschließen. Dies wird auch in der Literatur als Vorbereitung bezeichnet.

Da gut implementierte Kopositivitätstests sich dies stets zunutze machen, wollen wir hier einige dieser effizienten Tests vorstellen. Man beachte, dass sich jedes der folgenden Kriterien in höchstens polynomieller Zeit testen lässt.

Wir nennen im Folgenden einen Vektor $x \in \mathbb{R}^d$ verletzend, falls $Q[x] < 0$ ist. Viele der folgenden Kriterien sind notwendig für Kopositivität. Wir stellen hier aber nur eine Auswahl vor, da die „vollständige“ Liste allerdings entweder noch nicht existiert, sehr lang oder über zu viele Veröffentlichungen verteilt sein wird.

Wir fassen einige notwendige Bedingungen in einem Lemma zusammen, die alle durch einfache Rechnungen folgen.

Lemma 3.3. *Für eine kopositive Matrix Q gelten*

1. $q_{ii} \geq 0$. Andernfalls ist $x = e_i$ verletzend.
2. Ist $q_{ii} = 0$, so müssen alle Einträge in der i -ten Spalte beziehungsweise i -ten Zeile nichtnegativ sein. Sonst ist für ein $j \neq i$ mit $q_{ij} < 0$ der Vektor $x = (q_{jj} + 1)e_i - q_{ij}e_j$ verletzender.
3. $q_{ij} \geq -\sqrt{q_{ii}q_{jj}}$. Ist dies nicht der Fall so ist $x = \sqrt{q_{ii}}e_i + \sqrt{q_{jj}}e_j$ ein verletzender Vektor.
4. $q_{ij} \geq -\frac{1}{2}(q_{ii} + q_{jj})$, da sonst $x = e_i + e_j$ verletzend ist.
5. $\sum_{i,j=1}^d q_{ij} \geq 0$. Ansonsten ist $x = e$ verletzend.

Man beachte, dass jede algorithmisch realisierbare notwendige Bedingung für Kopositivität zur Vorbereitung eingesetzt werden kann, jedoch es nur dann sinnvoll ist, wenn der Aufwand des Tests recht „klein“ ist.

Die obigen Bedingungen können außerdem recht einfach für strikte Kopositivität angepasst werden. Nur Punkt 2 kann nicht benutzt werden, da dadurch Punkt 1 verletzt werden würde.

Für strikte Kopositivität ergibt sich allerdings ein neues Kriterium. Sollte sich im Kern der Matrix Q ein nichtnegativer Vektor x befinden, so können wir folgern, dass $Q \notin \text{int } \mathcal{COP}^d$ ist, da in diesem Fall $Qx = 0$ und damit $Q[x] = 0$ ist.

Mit (3.10) ist es eine sinnvolle Strategie die Matrix Q nach Zweierblöcken $\begin{pmatrix} q_{ii} & q_{ij} \\ q_{ij} & q_{jj} \end{pmatrix}$ mit $q_{ij} < 0$ zu durchsuchen. Sollte sich ergeben, dass es so einen Block gibt, der nicht kopositiv ist, dann können wir wie in Punkt 4 von Lemma 3.3 einen verletzenden Vektor konstruieren. Ähnlich könnte man auch mit den Kriterien für 3×3 - und 4×4 -Blöcke verfahren.

Ein anderer Teil der Vorbereitung ist das Reduzieren der Matrix, sowohl in Größe der Dimension als auch der Einträge.

Zunächst wissen wir durch (2.1), dass Q genau dann kopositiv ist, wenn DQD kopositiv ist, wobei D eine nichtnegative nichtsinguläre Diagonalmatrix ist. Die Einträge von Q lassen sich also in einem gewissen Sinne skalieren, wenn wir auf Kopositivität testen wollen.

Ist die Diagonale von Q beispielsweise positiv, so können wir Q mit der Diagonalmatrix $D = \text{diag}(1/\sqrt{q_{11}}, \dots, 1/\sqrt{q_{dd}})$ so skalieren, dass die Diagonale auf 1 „normiert“ wird. Man beachte, dass sich dadurch die Punkte 3 und 4 in Lemma 3.3 zu $q_{ij} \geq -1$ vereinfachen.

Tatsächlich lassen sich sogar alle positiven Außerdiagonaleinträge auf 1 „normieren“. Dafür betrachten wir die Matrizen

$$Q = \begin{pmatrix} 1 & b^T & q_{1d} \\ b & A & c \\ q_{1d} & c^T & 1 \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} 1 & b^T & 1 \\ b & A & c \\ 1 & c^T & 1 \end{pmatrix}.$$

Bomze und Eichfelder zeigten nun in [22, Lemma 4.4], dass Q genau dann kopositiv ist, wenn B kopositiv ist, indem sie aus einem verletzenden Vektor von B einen für Q konstruierten. Ist $x = \begin{pmatrix} x_1 & a^T & x_d \end{pmatrix}$ ein verletzender Vektor von B so ist der Vektor $\begin{pmatrix} \max\{0, -a^T b\} & a^T & 0 \end{pmatrix}$ verletzend für Q , falls $a^T b \leq a^T c$ ist. Andernfalls ist der Vektor $\begin{pmatrix} 0 & a^T & \max\{0, -a^T c\} \end{pmatrix}$ verletzend, wie man durch Nachrechnen prüft. Da Permutationen wie in (2.1) Kopositivität nicht beeinflussen, lässt sich dieser Prozess für alle positiven Diagonaleinträge durchführen. Eine kopositive Matrix hat nach diesen beiden Transformationen nur noch Einträge aus $[-1, 1]$, was sich einfach testen lässt.

Um die Größe der Matrix zu reduzieren benötigen wir zunächst, dass es eine nichtnegative Zeile beziehungsweise Spalte gibt. Angenommen, die i -te Zeile und Spalte der Matrix Q ist nichtnegativ. Dann genügt es nur Vektoren mit $x_i = 0$ zu betrachten, denn

$$Q[x] = Q[\tilde{x} + x_i e_i] = Q[\tilde{x}] + x_i \tilde{x}^T Q e_i + x_i^2 q_{ii}.$$

Hierbei ist $\tilde{x} = x - x_i e_i$. Da $Q e_i$ der i -ten Zeile entspricht ist der zweite und dritte Summand immer nicht negativ, solange $x_i \geq 0$ ist. Anders gesagt spielt die i -te Zeile beziehungsweise Spalte für die Kopositivität von Q keine Rolle. Wir können sie also „entfernen“ und erhalten eine $(d-1) \times (d-1)$ -Matrix \tilde{Q} .

Ist \tilde{Q} nicht kopositiv, so können wir aus einem verletzenden Vektor von \tilde{Q} einen verletzenden Vektor für Q konstruieren, indem wir an der i -ten Stelle eine 0 „hineinschieben“.

Kopositivitätskriterium von Gaddum

Mit der Vorbereitung aus dem Weg wollen wir nun den klassischen Kopositivitätstest, welcher auf Gaddum [55] zurückgeht, behandeln.

Im Folgenden bezeichnen wir den $d-1 \times d-1$ -Hauptminor einer symmetrischen Matrix Q , welcher durch Entfernen der i -ten Zeile und Spalte entstanden ist, mit Q^i .

Wie wir in der vorherigen Sektion angemerkt haben, spielt die Kopositivität der Hauptminoren einer Matrix eine gewisse Rolle für die Kopositivität der Matrix selber. Genauer folgt aus der Kopositivität einer Matrix Q die Kopositivität ihrer Hauptminoren. Damit ist dies zunächst ein notwendiges Kriterium für Kopositivität.

Gaddum beobachtete, dass dies zu einem hinreichendem Kriterium erweitert werden kann, indem man zusätzlich die Bedingung

$$\min_{x \in \Delta^S} \max_{y \in \Delta^S} x^T Q y^T = \min_{y \in \Delta^S} \max_{x \in \Delta^S} x^T Q y \geq 0 \quad (3.11)$$

betrachtet [55, Theorem 3.2]. Dabei ist (3.11) der Wert des sogenannten *Zwei-Spieler-Spiels mit Matrix Q* [55, Sektion 3].

Daraus ergibt sich nun ein rekursiver Kopositivitätstest, siehe Algorithmus 5.

```

Input : Matrix  $Q \in \mathcal{S}^d$ 
Output: „kopositiv“ oder „nicht kopositiv“
Wende Algorithmus 5 auf die  $d-1 \times d-1$ -Hauptminoren  $Q^i$  von  $Q$ 
if Es gibt ein  $i$ , sodass  $Q^i$  nicht kopositiv ist then
  | Gib „nicht kopositiv“ aus
end
Berechne  $\text{val} = (3.11)$ 
if  $\text{val} \geq 0$  then
  | Gib „kopositiv“ aus
else
  | Gib „nicht kopositiv“ aus
end

```

Algorithmus 5: Rekursiver Kopositivitätstest, basierend auf Gaddums Kriterium

Die hier interessante Eigenschaft von (3.11) ist, dass sich der Wert über das lineare Optimierungsproblem [65, Sektion 3.1]

$$\begin{aligned} & \max y_1 - y_2 \\ & \text{so, dass } Qx - y_1 e + y_2 e \leq 0, \\ & \quad y_1, y_2 \geq 0 \\ & \text{und } x \in \Delta^S \end{aligned}$$

berechnen lässt.

Man beachte, dass Algorithmus 5 trotzdem keinen polynomiellen Aufwand hat. Genauer bezeichne $P(n)$ den Aufwand zum Lösen von (3.11). Dann ist der Aufwand von Algorithmus 5 $\sum_{n=1}^d \binom{d}{n} P(n)$ [65, Sektion 3.1]. Damit ist Gaddums Test nur für recht kleine Dimension geeignet.

Für uns wichtig ist, dass wir Gaddums Kopositivitätstest zu einem strikten Kopositivitätstest machen können, indem wir fordern, dass die Hauptminoren der Matrix Q ebenfalls strikt kopositiv sein sollen und der Wert (3.11) ebenfalls positiv sein soll [55, Theorem 3.1].

Kopositivitätstests durch gemischt ganzzahlige Optimierungsprobleme

Obwohl Gaddums Kriterium ein klassischer Kopositivitätstest ist, sehen wir, dass der Aufwand dieses Tests selbst für kleine Dimension recht groß werden kann. Da wir wissen, dass wir lineare Optimierungsprobleme in polynomieller Zeit lösen können ist die große Schwierigkeit in Algorithmus 5 die Rekursion.

Es stellt sich also die Frage, ob wir auf die Rekursion verzichten können, wenn wir im Gegenzug dazu schwierigere Optimierungsprobleme lösen. Nie, Yang und Zhang geben in [98] beispielsweise einen auf sogenannten „semidefiniten Problemen“, konische Probleme über $\mathcal{S}_{\geq 0}^d$, basierenden Algorithmus an, welcher keinerlei Rekursion beinhaltet. Ebenso geben Bomze und Eichfelder in [22] einen Algorithmus der lineare Optimierungsprobleme und sogenannte „Simplexpartitionen“ verwendet. Die Schwierigkeit hier liegt bei diesen Simplexpartitionen, wie wir in Kapitel 5 sehen werden.

Kürzlich sind Tests basierend auf gemischt ganzzahligen Optimierungsproblemen aufgetaucht [2][6][120], welche nach ersten Erfahrungen wohl recht effizient sein sollen. Wir wollen hier einen solchen Test als Alternative vorstellen.

Wie wir bereits in Sektion 3.2.3 angemerkt haben, ist ein ausreichendes Kriterium für die Kopositivität von Q , dass $Q[x]$ nichtnegativ auf dem Standardsimplex Δ^S ist. Dies liegt daran, dass jeder nichtnegative Vektor auf einem Strahl liegt, welcher von einem Vektor in Δ^S erzeugt wird.

Diese Bedingung können wir über das im Allgemeinen nicht-konvexe Standard-Quadratische-Problem

$$\min \{Q[x] \mid x \in \Delta^S\} \geq 0 \quad (3.12)$$

ausdrücken (siehe auch Sektion 3.2.3). Xia, Vera und Zuluaga konnten kürzlich in [120, Sektion 2] (siehe auch [6, Sektion 2.1]) zeigen, dass man (3.12) in das gemischt ganzzahlige lineare Optimierungsproblem

$$\begin{aligned} & \min -\mu, \\ & \text{so dass } Qx + \mu e - \nu = 0, \\ & e^T x = 1, 0 \leq x \leq y, \\ & 0 \leq \nu \leq 2d \max_{i,j} |q_{ij}|(e - y), \\ & \text{und } y \in \{0, 1\}^d \end{aligned} \quad (3.13)$$

umformulieren kann. Damit ist Q genau dann kopositiv, wenn der optimale Wert von (3.13) nichtnegativ ist. Im Gegensatz zu (3.12) ist (3.13) immer konvex und lässt sich daher im Allgemeinen „besser“ lösen. Den Autoren von [6] zufolge ist diese Art von Test bereits für 6×6 -Matrizen besser als Algorithmus 5 und soll wohl um Einiges besser skalieren.

4 Ein möglicher Algorithmus zum Lösen des \mathcal{CP} -Membership-Problems

Wie schon in Sektion 3.2.3 bezeichnen wir mit dem \mathcal{CP} -Membership-Problem oder Strong Membership Problem des Kegels \mathcal{CP}^d das Entscheidungsproblem „Gegeben eine rationale Matrix Q , ist Q vollständig positiv?“.

Im folgenden Kapitel wollen wir uns mit diesem Problem beschäftigen.

4.1 Das \mathcal{CP} -Membership-Problem

In Sektion 3.2.3 haben wir angemerkt, dass das \mathcal{COP} -Membership-Problem \mathcal{NP} -vollständig ist. Da \mathcal{CP}^d und \mathcal{COP}^d dual zueinander sind ist anzunehmen, dass das \mathcal{CP} -Membership-Problem eine ähnliche Komplexität hat.

Dies wurde lange vermutet, konnte jedoch erst kürzlich durch Dickinson und Gijben in [43] bewiesen werden. Dort wurde gezeigt, dass das \mathcal{CP} -Membership-Problem \mathcal{NP} -schwer ist. Dazu zeigten sie, dass sich dieses Problem essentiell auf das, schon in Sektion 2.3.2 erwähnte INDSET, reduzieren lässt [43, Theorem 5.3]. Ob das \mathcal{CP} -Membership-Problem selber in \mathcal{NP} liegt, ist bisher noch unbekannt.

Allerdings gibt es eine Vielzahl an verschiedenen Algorithmen die versuchen, dieses Problem wenigstens teilweise zu lösen. Diese lassen sich grob in zwei Klassen unterteilen. Einerseits gibt es numerische Methoden, die recht effizient approximative \mathcal{CP} -Faktorisierungen berechnen können. Man beachte, dass eine \mathcal{CP} -Faktorisierung von Q ein Zertifikat für $Q \in \mathcal{CP}^d$ ist. Algorithmen, die in diese Klasse fallen, finden sich in [51][56] [69][97][117]. Die andere Klasse von Algorithmen sind theoretische Algorithmen, die zwar exakte \mathcal{CP} -Faktorisierungen berechnen können, aber entweder nur für bestimmte Klassen von Eingaben oder nicht unbedingt terminieren. Beispiele für solche Algorithmen finden sich in [6][13][41, Sektion 3.3][42][47]. Letzter ist der Algorithmus, den wir in der folgenden Sektion vorstellen wollen.

4.2 Modifikation von Algorithmus 1

In [47] stellen die Autoren eine einfache Modifikation von Algorithmus 1 zum Lösen des \mathcal{CP} -Membership-Problems vor, siehe Algorithmus 6.

Input : rationale Matrix $Q \in \mathcal{S}^d$
Output: Falls $Q \in \widetilde{\mathcal{CP}}^d$, dann eine rationale \mathcal{CP} -Faktorisierung von A .
 Falls $Q \notin \mathcal{CP}^d$, dann einen Zeugen $W \in \mathcal{COP}^d$ mit $\langle Q, W \rangle < 0$
 Wähle eine erste \mathcal{COP} -perfekte Matrix $P = P_{\text{Start}} \in \mathcal{R}$. Initialisiere $\mathcal{V}(P)$.
while $Q \notin \mathcal{V}(P)$ **do**
 if $\langle P, Q \rangle < 0$ **then**
 | Gib $Q \notin \mathcal{CP}^d$ und $W = P$ aus
 end
 Bestimme einen Erzeuger R eines Extremalstrahles von $\mathcal{V}(P)^*$ mit
 $\langle R, Q \rangle < 0$.
 if $R \in \mathcal{COP}^d$ **then**
 | Gib $Q \notin \mathcal{CP}^d$ und $W = R$ aus
 end
 Bestimme benachbarte \mathcal{COP} -perfekte Matrix N mit Algorithmus 4
 Setze $P = N$
end
 Bestimme $\alpha_1, \dots, \alpha_n \in \mathbb{Q}_{\geq 0}$, sodass $Q = \sum_{i=1}^n \alpha_i v_i v_i^T$
 Gib $Q \in \widetilde{\mathcal{CP}}^d$ und Faktorisierung aus

Algorithmus 6: Algorithmus zum Lösen des \mathcal{CP} -Membership-Problems und Berechnen einer rationalen \mathcal{CP} -Faktorisierung, siehe [47, Prozedur 3]

Mit der \mathcal{CP} -Faktorisierung, die essentiell von Algorithmus 1 produziert wird, haben wir ein Zertifikat für $Q \in \widetilde{\mathcal{CP}}^d$. Es fehlt also nur noch eines für $Q \notin \mathcal{CP}^d$. Algorithmus 6 benutzt ein recht natürliches (im Allgemeinen nicht algorithmisch berechenbares) Zertifikat für $Q \notin \mathcal{CP}^d$. Mit (2.2) wissen wir, wenn es eine kopositive Matrix B mit $\langle B, Q \rangle < 0$ gibt, kann Q nicht vollständig positiv sein. Dabei betrachtet Algorithmus 6 nur solche kopositive Matrizen, die entweder \mathcal{COP} -perfekt sind oder als Extremalstrahl eines Kegels $(\mathcal{V}(P))^*$ auftreten. Wie wir gleich sehen werden, sind diese Extremalstrahlen gerade die nichtnegativen Extremalstrahlen E_{ij} , $i \neq j$ von \mathcal{COP}^d .

Es bleibt also zunächst zu sehen, dass wir tatsächlich jede nicht vollständig positive durch \mathcal{COP} -perfekte Matrizen und die nichtnegativen Extremalstrahlen erfassen können.

Die generelle Idee dafür ergibt sich aus folgender Darstellung von \mathcal{CP}^d . Es ist [47, Theorem 2.3]

$$\mathcal{CP}^d = \left\{ Q \in \mathcal{S}^d \mid \langle Q, B \rangle \geq 0 \text{ für alle Ecken und Extremstrahlen } B \text{ von } \mathcal{R} \right\}. \quad (4.1)$$

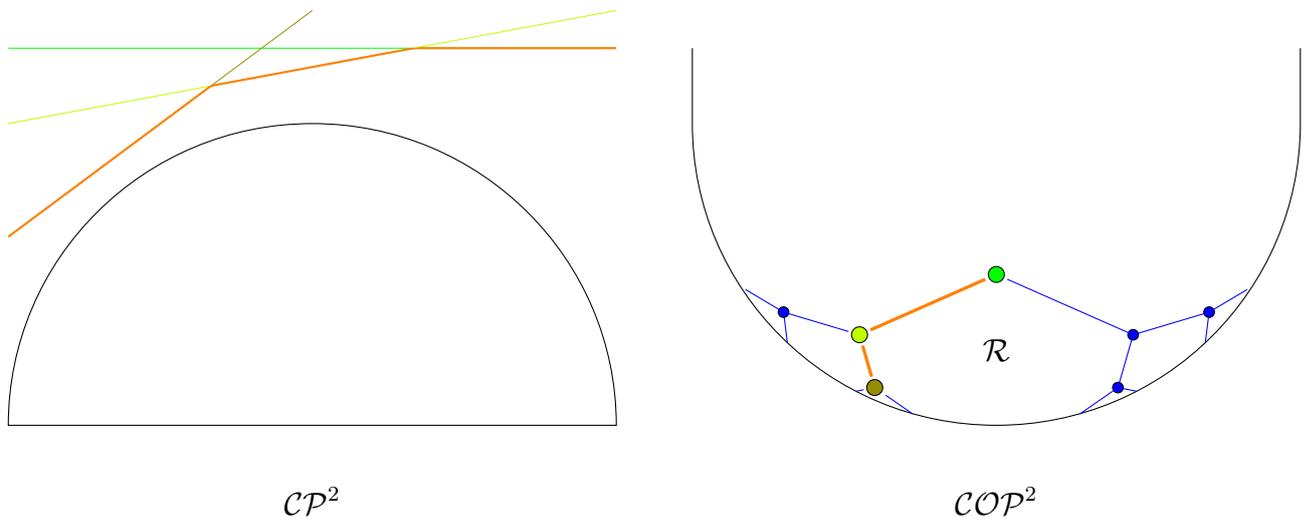
Man beachte, dass die Ecken von \mathcal{R} gerade die \mathcal{COP} -perfekten Matrizen mit einem

kopositiven Minimum von 1 sind. Diese Darstellung folgt wie in der Argumentation in [47, Theorem 2.3] mit (2.2), (2.29), [10, Theorem 2.3.1 (iv)] und [46, Lemma 2.4]. Zunächst ist mit (2.29) $\text{int } \mathcal{COP}^d = \text{cone } \mathcal{R} \setminus \{0\}$. Weiter ist mit [10, Theorem 2.3.1 (iv)] und (2.2) $(\text{int } \mathcal{COP}^d)^* = (\mathcal{COP}^d)^* = \mathcal{CP}^d$. Der Rest folgt nun aus der Tatsache, dass \mathcal{R} ein lokal endlicher Polyeder ist, also lokal wie ein Polytop aussieht, was in der obigen Darstellung resultiert.

Geometrisch gesehen konstruiert Algorithmus 6 also sukzessiv durch

$$\mathcal{CP}^d \subseteq \mathcal{CP}_n = \{A \in \mathcal{S}^d \mid \langle Q, B \rangle \geq 0 \text{ für alle, bis zum Schritt } n \text{ des Algorithmus, konstruierten } \mathcal{COP}\text{-perfekten Matrizen } B\}$$

eine immer enger werdende Approximation an \mathcal{CP}^d .



Es bleiben also die Extremalstrahlen von \mathcal{R} zu klassifizieren. Man beachte dazu, dass die Extremalstrahlen von \mathcal{R} die Form $P + aR$, $a \geq 0$ für ein geeignetes $R \in \mathcal{COP}^d$ und \mathcal{COP} -perfektes P haben. Damit erhalten wir folgendes

Theorem 4.1. *Die Extremalstrahlen von \mathcal{R} haben die Form $P + aE_{ij}$, $i \neq j$ für eine \mathcal{COP} -perfekte Matrix P . Insbesondere sind daher die möglichen kopositiven Extremalstrahlen der Kegel $\mathcal{V}(P)^*$ von der Form E_{ij} , $i \neq j$.*

Beweis. Wir zeigen hier nur die Behauptung für \mathcal{R} , der zweite Teil folgt analog mit (3.4).

Man beachte zunächst, dass die Extremalstrahlen 1-Seiten von \mathcal{R} sind. Sei nun eine \mathcal{COP} -perfekte Matrix P mit $\min_{\mathcal{COP}} P = 1$ fest gewählt. Damit wir eine 1-Seite von \mathcal{R} erhalten, die P enthält, muss $\langle R, vv^T \rangle = 0$ für $\binom{d+1}{2} - 1$ linear unabhängige vv^T mit $v \in \text{Min}_{\mathcal{COP}} P$ sein.

Da R , nach Voraussetzung, kopositiv ist, induziert R mit

$$F = \{Q \in \mathcal{CP}^d \mid \langle R, Q \rangle = 0\}$$

eine Seite von \mathcal{CP}^d (siehe auch Sektion 2.4.3).

Nach obiger Konstruktion enthält F also $\binom{d+1}{2} - 1$ linear unabhängige Elemente von \mathcal{CP}^d . Daher ist F offenbar eine Facette. Mit Lemma 2.2 erhalten wir dann, dass F , wie oben, durch ein E_{ij} , $i \neq j$ induziert wird. Somit ist $R = E_{ij}$. \square

Eine sofortige Folgerung von Theorem 4.1 ist, dass wir (4.1) vereinfachen können. Es ergibt sich

$$\mathcal{CP}^d = \left\{ Q \in \mathcal{S}^d \mid \langle B, Q \rangle \geq 0 \text{ für alle } \mathcal{COP}\text{-perfekten Matrizen } B \text{ und } \langle E_{ij}, Q \rangle \geq 0 \text{ für } i \neq j \right\}.$$

Man beachte, dass es daher bis auf Symmetrie nur eine „Richtung ins Unendliche“ in \mathcal{R} gibt, siehe Sektion 2.4.

Weiter gibt uns Theorem 4.1 zusammen mit (4.1), dass wir tatsächlich alle nicht vollständig positiven durch die \mathcal{COP} -perfekten Matrizen und die nichtnegativen Extremalstrahlen E_{ij} , $i \neq j$ erfassen können. Algorithmus 6 gibt uns also bei Termination die gewünschte Ausgabe.

Theorem 4.1 erlaubt es uns auch Algorithmus 6 auf zwei Arten zu modifizieren. Zum einen reduziert sich der Kopositivitätstest in der **while**-Schleife von Algorithmus 6 zu einem Nichtnegativitätstest, der sich offenbar in linearer Zeit durchführen lässt. Diese Modifikation belässt Algorithmus 6 in seiner ursprünglichen Form. Man beachte hier, dass die nicht vollständig positiven Matrizen mit einem außerdiagonalen negativen Eintrag sowohl durch einen Extremalstrahl als auch durch eine \mathcal{COP} -perfekte Matrix erfasst werden können. Solche, die negative Einträge auf der Diagonalen haben, aber keine außerdiagonalen, allerdings nur durch \mathcal{COP} -perfekte Matrizen erfasst werden.

Die zweite mögliche Modifikation ist, einen Nichtnegativitätstest der Eingabematrix an den Anfang des Algorithmus zu schieben und daher auf den Kopositivitätstest des Extremalstrahls R in der **while**-Schleife von Algorithmus 6 zu verzichten. Hierbei würde auch nur ein Nichtnegativitätstest der Außerdiagonalelemente der Eingabematrix ausreichen. Mit dieser Modifikation ergibt sich Algorithmus 7.

Wir wissen nun, dass Algorithmus 6 und 7 bei Termination das gewünschte Ergebnis liefern. Im Gegensatz zum („Kern“-)Algorithmus 1 ist es allerdings noch offen, ob sie dies immer tun. Für eine Eingabematrix aus $\widetilde{\mathcal{CP}}^d$ ist dies mit Theorem 3.1 sicherlich der Fall. Für eine nicht vollständig positive Eingabematrix besteht folgende offene

Vermutung 4.2. [47, Vermutung 3.2] *Für eine rationale Matrix $Q \notin \mathcal{CP}^d$ terminiert Algorithmus 6 (Algorithmus 7), mit geeigneter Pivotregel, nach endlichen vielen Schritten.*

Generell haben wir mit den Erfahrungen in [47, Sektion 5] guten Grund zu der Annahme, dass Vermutung 4.2 wahr ist. Wir werden in der nächsten Sektion zeigen, dass dies für $d = 2$ tatsächlich der Fall ist und betrachten in Sektion 4.4 die Fälle $d = 3$ und $d = 4$ etwas näher.

Weiter lässt sich sagen, dass wenn Vermutung 4.2 und [115, Vermutung 1] (siehe auch Kapitel 3) wahr sind, löst Algorithmus 6 das \mathcal{CP} -Membership-Problem.

Input : rationale Matrix $Q \in \mathcal{S}^d$

Output: Falls $Q \in \widetilde{\mathcal{CP}}^d$, dann eine rationale \mathcal{CP} -Faktorisierung von A .
 Falls $Q \notin \mathcal{CP}^d$, dann einen Zeugen $W \in \mathcal{COP}^d$ mit $\langle Q, W \rangle < 0$

if $Q \notin \mathcal{N}^d$ **then**
 | Gib $Q \notin \mathcal{CP}^d$ und $W = E_{ij}$ für passende i, j aus
end

Wähle eine erste \mathcal{COP} -perfekte Matrix $P = P_{\text{Start}} \in \mathcal{R}$. Initialisiere $\mathcal{V}(P)$.

while $Q \notin \mathcal{V}(P)$ **do**
 | **if** $\langle P, Q \rangle < 0$ **then**
 | | Gib $Q \notin \mathcal{CP}^d$ und $W = P$ aus
 | **end**
 | Bestimme einen Erzeuger R eines Extremalstrahles von $\mathcal{V}(P)^*$ mit
 | $\langle R, Q \rangle < 0$.
 | Bestimme benachbarte \mathcal{COP} -perfekte Matrix N mit Algorithmus 4
 | Setze $P = N$
end

Bestimme $\alpha_1, \dots, \alpha_n \in \mathbb{Q}_{\geq 0}$, sodass $Q = \sum_{i=1}^n \alpha_i v_i v_i^T$

Gib $Q \in \widetilde{\mathcal{CP}}^d$ und Faktorisierung aus

Algorithmus 7: Modifizierter Algorithmus 6, basierend auf Theorem 4.1

Zu [115, Vermutung 1] lässt sich sagen, dass wegen [46, Theorem 1.1] nur noch der Rand von \mathcal{CP}^d zu betrachten ist. Da dieser, wie wir Sektion 2.4 angemerkt haben, recht schlecht verstanden ist, ist zu erwarten, dass [115, Vermutung 1] nicht in nächster Zeit gelöst wird. Für eine ausführliche Übersicht des diesbezüglichen Wissensstandes verweisen wir auf [15][115].

4.3 Termination in Dimension 2

In dieser Sektion wollen wir uns mit dem Fall $d = 2$ von Vermutung 4.2 beschäftigen. Dabei beziehen wir uns der Einfachheit halber nur auf Algorithmus 6, die Aussagen gelten aber analog auch für Algorithmus 7. Genauer beweisen wir das folgende

Theorem 4.3. *Für eine nicht vollständig positive rationale $Q = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ terminiert Algorithmus 6, mit geeigneter Pivotregel, nach endlich vielen Schritten.*

Zunächst wollen wir zwei nützliche Eigenschaften des 2×2 -Fall besprechen, welche wir für den Beweis von Theorem 4.3 benutzen werden.

Ein allgemeine Beobachtung \mathcal{COP} -perfekter Matrizen ist, dass sie nicht in \mathcal{N}^d , $d > 1$ liegen können. Man beachte dafür, dass das kopositive Minimum einer Matrix höchstens so groß sein kann, wie der kleinste Diagonaleintrag dieser Matrix. Wäre also eine \mathcal{COP} -perfekte Matrix nichtnegativ, so ist $\text{Min}_{\mathcal{COP}} P \subseteq \{e_1, \dots, e_d\}$. Dies sind offenbar nicht genügend minimale Vektoren, um Definition 2.3 zu genügen, sodass dies einen Widerspruch ergibt.

Im 2×2 -Fall erhalten wir also, dass der Außerdiagonaleintrag p_{12} einer $2 \times \mathcal{COP}$ -perfekten Matrix P negativ sein muss. Mit Lemma 3.2 erhalten wir daher, dass P positiv definit ist. Wie wir in der nächsten Sektion sehen werden, gilt dies nur im 2×2 -Fall. Insbesondere folgt mit Definition 2.3, dass jede 2×2 \mathcal{COP} -perfekte Matrix P auch eine perfekte quadratische Form ist und daher auch $|\text{Min}_{\mathcal{COP}} P| = 3$ gilt [79].

Für die andere Eigenschaft beobachten wir zunächst, dass mit der obigen Eigenschaft die Kegel $(\mathcal{V}(P))^*$ genau 3 Extremalstrahlen haben. Mit Abbildung 3.2 sehen wir, dass daher in jedem Schritt von Algorithmus 6 höchstens 2 mögliche Extremalstrahlen zur Auswahl stehen.

Wenn man in Algorithmus 6 bei $\frac{1}{2}Q_{A_2}$ anfängt, kann man für einen gewählten Extremstrahl R folgende Formel verwenden, um die benachbarte \mathcal{COP} -perfekte Matrix N zu berechnen. Wir wollen diese Formel hier nicht herleiten, sondern verweisen stattdessen auf [59][67]:

Sei P eine 2×2 \mathcal{COP} -perfekte Matrix mit $\text{Min}_{\mathcal{COP}} P = \left\{ \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix}, \begin{pmatrix} e \\ f \end{pmatrix} \right\}$. Ist der gewählte Extremalstrahl R so, dass $R \left[\begin{pmatrix} e \\ f \end{pmatrix} \right] > 0$ ist, so ist die im Algorithmus 6 berechnete benachbarte \mathcal{COP} -perfekte Matrix N gegeben durch [47, Sektion 4.1]

$$N = P + 2 \begin{pmatrix} bd & -\frac{1}{2}(ad + bc) \\ -\frac{1}{2}(ad + bc) & ac \end{pmatrix}. \quad (4.2)$$

Des weiteren ist $\text{Min}_{\mathcal{COP}} N = \left\{ \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix}, \begin{pmatrix} a+c \\ b+d \end{pmatrix} \right\}$.

Dies brauchen wir nur für den Fall $P = \frac{1}{2}Q_{A_2}$ nachweisen, da mit [79] für jede \mathcal{COP} -perfekte Matrix P eine ganzzahlig unimodulare Matrix $U \in \mathbf{GL}_2(\mathbb{Z})$ existiert, sodass $P = U^T \frac{1}{2}Q_{A_2} U$. Diese Eigenschaft wird auch „arithmetische Äquivalenz“ genannt, siehe auch [111, Sektion 1.1.2]. Insbesondere gilt damit auch, dass $\mathcal{V}(P) = U^T \mathcal{V}(\frac{1}{2}Q_{A_2}) U$ ist.

Wir können also den Kegel $\mathcal{V}(P)$ und damit auch die Extremalstrahlen von $(\mathcal{V}(P))^*$ so transformieren, dass sie $\mathcal{V}(\frac{1}{2}Q_{A_2})$ und den zugehörigen Extremalstrahlen entsprechen. Es genügt also (4.2) für diesen Spezialfall zu betrachten. Für diesen folgt (4.2) mit einfachen Rechnungen.

Durch die obige Transformation wird offenbar immer einer der 3 Extremalstrahlen von $(\mathcal{V}(P))^*$ auf den kopositiven Extremalstrahl E_{12} von $(\mathcal{V}(P))^*$ abgebildet und ist damit eine „unmögliche Richtung“. Tatsächlich wird immer genau der Extremalstrahl auf E_{12} transformiert, der zu $\frac{1}{2}Q_{A_2}$ „zurückführen“ würde. Daher ist die Formel nur für den „Hinweg“ anwendbar.

Wir zeigen nun zunächst einen Spezialfall von Theorem 4.3. Es gilt das

Lemma 4.4. *Für eine nicht vollständig positive rationale Matrix $Q = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \notin \mathcal{CP}^2$ mit $b \geq 0$ terminiert Algorithmus 6 mit $P_{\text{Start}} = \frac{1}{2}Q_{A_2}$ nach endlich vielen Schritten.*

Beweis. Da zunächst nach Voraussetzung Q rational ist, reicht es aus den Fall $a, b, c \in \mathbb{Z}$ zu behandeln, denn wenn Q nicht vollständig positiv ist, so trifft auch der von Q erzeugte Strahl \mathcal{CP}^2 nicht.

Nun ist $\langle \frac{1}{2}Q_{A_2}, Q \rangle = a + c - b$ ganzzahlig. Induktiv erhalten wir dann mit (4.2), dass

$$\langle N, Q \rangle = \langle P, Q \rangle + \left\langle 2 \begin{pmatrix} v_2 u_2 & -\frac{v_1 u_2 + v_2 u_1}{2} \\ -\frac{v_1 u_2 + v_2 u_1}{2} & v_1 u_1 \end{pmatrix}, Q \right\rangle \text{ ganzzahlig ist,}$$

wobei $(v_1, v_2)^T, (u_1, u_2)^T \in \text{Min}_{\mathcal{COP}} P \subset \mathbb{Z}^2$ sind. Da der Extremalstrahl R in Algorithmus 6 so gewählt wird, dass $\langle R, Q \rangle < 0$ ist, folgt dass $\langle N, Q \rangle < \langle P, Q \rangle$. Da beide Ausdrücke ganzzahlig sind, folgt die Behauptung. \square

Falls $b < 0$ ist, so terminiert Algorithmus 6 für $P_{\text{Start}} = \frac{1}{2}Q_{A_2}$ bereits im ersten Schritt, da E_{12} ein Extremalstrahl von $(\mathcal{V}(\frac{1}{2}Q_{A_2}))^*$ ist. Man beachte, dass $\frac{1}{2}Q_{A_2}$ die einzige 2×2 \mathcal{COP} -perfekte Matrix ist, die diesen Extremalstrahl „besitzt“, siehe auch Abbildung 3.2. Damit ist $\frac{1}{2}Q_{A_2}$ auch die einzige 2×2 \mathcal{COP} -perfekte Matrix, bei der Algorithmus 6 terminieren kann, falls die Eingabematrix eine positiv semidefinite Matrix mit negativen Außerdiagonaleinträgen ist.

Wählen wir also $\frac{1}{2}Q_{A_2}$ als Startpunkt in Algorithmus 6, so terminiert dieser mit geeigneter Pivotregel nach endlich vielen Schritten. Dies nutzen wir nun aus um Theorem 4.3 zu zeigen.

Wie in Lemma 4.4 reicht es aus den Fall $a, b, c \in \mathbb{Z}$ zu betrachten. Sei P nun eine \mathcal{COP} -perfekte Matrix. Mit [79] existiert wie vorhin eine ganzzahlige unimodulare Ma-

trix $U = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}$ so, dass $P = U^T \frac{1}{2} Q_{A_2} U$. Damit erhalten wir

$$\begin{aligned} \langle P, Q \rangle &= \left\langle U^T \frac{1}{2} Q_{A_2} U, Q \right\rangle \\ &= \left\langle \begin{pmatrix} (u_{11} - u_{21})^2 & u_{11}u_{12} + u_{22}u_{21} - \frac{u_{11}u_{22} + u_{12}u_{21}}{2} \\ u_{11}u_{12} + u_{22}u_{21} - \frac{u_{11}u_{22} + u_{12}u_{21}}{2} & (u_{12} - u_{22})^2 \end{pmatrix}, Q \right\rangle \\ &= (u_{11} - u_{21})^2 a + (u_{12} - u_{22})^2 c + (2(u_{11}u_{12} + u_{22}u_{21}) - u_{11}u_{22} + u_{12}u_{21}) b \in \mathbb{Z}. \end{aligned}$$

Theorem 4.3 folgt nun, da für die \mathcal{COP} -perfekten Matrizen P und N in Algorithmus 6 gilt, dass

$$\langle N, Q \rangle = \langle P + \lambda R, Q \rangle < \langle P, Q \rangle$$

ist, da $\langle R, Q \rangle < 0$ gilt, und beide Ausdrücke, wie oben gezeigt, ganzzahlig sind. \square

Wir haben in dieser Sektion ausschließlich rationale Matrizen betrachtet, da dies die einzigen sind, die in Computeranwendungen auftreten können. Die Aussage folgt für reelle nicht vollständig positive Matrizen Q mit einem ähnlichen Argument wie in Theorem 3.1, indem man rationale Matrizen in Umgebungen von Q betrachtet und Theorem 4.3 anwendet. Damit ist Vermutung 4.2 zumindest für den Fall $d = 2$ wahr.

4.4 3×3 und 4×4 \mathcal{COP} -perfekte Matrizen

Wir wollen uns nun mit dem Fall $d > 2$ von Vermutung 4.2 beschäftigen. Dafür betrachten wir die 3×3 und 4×4 \mathcal{COP} -perfekten Matrizen etwas genauer.

Zunächst ist die Eigenschaft, dass \mathcal{COP} -perfekte Matrizen auch perfekte quadratische Formen sind, ein Phänomen des 2×2 -Falles, da bereits eine Dimension höher \mathcal{COP} -perfekte Matrizen auftreten, die nur positiv semidefinit oder sogar indefinit sind. Beispielsweise ist die \mathcal{COP} -perfekte Matrix

$$P = \begin{pmatrix} 3 & 2 & -9/2 \\ 2 & 1 & -5/2 \\ -9/2 & -5/2 & 1 \end{pmatrix},$$

mit $\text{Min}_{\mathcal{COP}} P = \{e_2, e_1 + e_3, 2e_1 + e_3, 3e_1 + 2e_3, 2e_2 + e_3, 3e_2 + e_3, e\}$, indefinit.

Wie wir schon in der vorherigen Sektion festgestellt haben, muss eine \mathcal{COP} -perfekte Matrix einen negativen Eintrag enthalten. Lemma 3.2 gibt uns dann, dass daher wenigstens ein Hauptminor einer 3×3 \mathcal{COP} -perfekten Matrix positiv definit ist. Wir vermuten, dass es sogar tatsächlich immer wenigstens 2 sind. Dies könnte sich für einen Beweis von Vermutung 4.2 für $d = 3$ als hilfreich erweisen.

Der andere wichtige Teil unseres Beweises von Vermutung 4.2 für $d = 2$ war die Formel (4.2) zur Berechnung von benachbarten \mathcal{COP} -perfekten Matrizen. Für $d \geq 3$ ist noch keine solche Formel (4.2) bekannt. Ein Problem, welches eine solche Formel unmöglich machen könnte, gibt uns die folgende

Vermutung 4.5. Für $d \geq 3$ ist die maximale Größe von $\text{Min}_{\mathcal{COP}} P$, über alle $d \times d$ \mathcal{COP} -perfekten Matrizen P , unbeschränkt. Genauer vermuten wir, dass jede \mathcal{COP} -perfekte Matrix mit einer \mathcal{COP} -perfekten Matrix benachbart ist, die mehr minimale Vektoren besitzt.

Man beachte, dass für die perfekten quadratischen Formen gerade das Gegenteil des ersten Parts von Vermutung 4.5 gilt. Dies liegt daran, dass es bis auf arithmetische Äquivalenz nur endlich viele perfekte quadratische Formen mit arithmetischem Minimum 1 gibt [111, Theorem 3.4] und, wie wir in der vorherigen Sektion anmerkten, arithmetisch äquivalente Formen gleich viele minimale Vektoren besitzen.

Die im Sinne von Vermutung 4.5 „größte“ 3×3 \mathcal{COP} -perfekte Matrix, die wir bisher finden konnten, ist

$$\begin{pmatrix} 829 & 829 & -3065/2 \\ 829 & 829 & -3065/2 \\ -3065/2 & -3065/2 & 2833 \end{pmatrix},$$

mit 125 minimalen Vektoren. Dies bildet einen starken Kontrast zu den 3×3 perfekten quadratischen Formen, die alle 12 minimale Vektoren haben [111, Kapitel 3, Tabelle 1].

Wir haben nun schon gewisse Schwierigkeiten im 3×3 -Fall gesehen. Für den 4×4 -Fall wollen wir nun eine Vermutung aufstellen, welche es ermöglichen würde, diesen Fall zumindest teilweise auf den 3×3 -Fall zurückzuführen. Es besteht die

Vermutung 4.6. Sei P eine 4×4 \mathcal{COP} -perfekte Matrix mit $\text{min}_{\mathcal{COP}} P = 1$. Ist der Hauptminor P^4 eine 3×3 \mathcal{COP} -perfekte Matrix mit $\text{min}_{\mathcal{COP}} P = 1$, so gibt es für jeden Nachbar N von P^4 einen Nachbar N' von P , sodass $N'^4 = N$ ist.

Man beachte hier, dass Vermutung 4.6 wegen (2.1) für einen beliebigen Hauptminor einer \mathcal{COP} -perfekten Matrix P formuliert werden kann. Das liegt daran, dass Transformationen der Form $P' = U^T P U$ für eine Permutationsmatrix U \mathcal{COP} -Perfektheit erhalten, da offenbar $\text{Min}_{\mathcal{COP}} P' = U \text{Min}_{\mathcal{COP}} P$ ist.

Weiter glauben wir, dass wenn Vermutung 4.6 wahr ist, eine analoge Version für $d \geq 5$ gilt. Für $d = 3$ gilt eine solche Aussage nicht. Dies liegt vermutlich daran, dass der Fall $d = 2$, wie oben schon ausgeführt, sehr speziell ist.

Vermutung 4.6 und etwaige Verallgemeinerungen würde es erlauben, Vermutung 4.2 zu einem großen Teil induktiv zu beweisen.

Schlussendlich wollen wir noch auf eine „Besonderheit“ der Nachbarschaft von $\frac{1}{2}Q_{A_d}$, $d = 3, 4$, im Bezug auf Algorithmus 6 und die in Sektion 4.2 besprochenen Modifikationen, aufmerksam machen.

Wie wir bereits in Sektion 2 angemerkt haben, ist E_{12} ein Extremalstrahl von $(\mathcal{V}(\frac{1}{2}Q_{A_2}))^*$. Allgemeiner ist E_{1d} ein Extremalstrahl von $(\mathcal{V}(\frac{1}{2}Q_{A_2}))^*$ wie man einfach mit (3.4) durch Wahl von $a_i^* = e$, sieht. Damit terminiert Algorithmus 6, wie auch schon für $d = 2$, mit Startpunkt $\frac{1}{2}Q_{A_d}$ bereits im ersten Schritt, falls $q_{1d} < 0$ ist.

Für $d = 3$ kommen E_{12} und E_{13} schon als Extremalstrahlen der Kegel $(\mathcal{V}(P))^*$ für

zwei Nachbarn von $\frac{1}{2}Q_{A_3}$

$$\begin{pmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \\ -1/2 & -1/2 & 1 \end{pmatrix} \text{ und } \begin{pmatrix} 1 & -1/2 & -1/2 \\ -1/2 & 1 & 0 \\ -1/2 & 0 & 1 \end{pmatrix}$$

vor. Man beachte, dass die obigen Matrizen permutierte Versionen von $\frac{1}{2}Q_{A_3}$ sind.

Für $d = 4$ gibt es fünf weitere Nichtnegative Extremalstrahlen, $E_{12}, E_{13}, E_{23}, E_{24}$ und E_{34} . Diese treten beispielsweise als Extremalstrahlen der Kegel $(\mathcal{V}(P))^*$ der Nachbarn

$$\begin{pmatrix} 1 & 0 & -1/2 & 0 \\ 0 & 1 & -1/2 & 0 \\ -1/2 & -1/2 & 1 & -1/2 \\ 0 & 0 & -1/2 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1/2 & 0 & 0 \\ -1/2 & 1 & -1/2 & -1/2 \\ 0 & -1/2 & 1 & 0 \\ 0 & -1/2 & 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & -1/2 & 0 & -1/2 \\ -1/2 & 1 & -1/2 & 1/2 \\ 0 & -1/2 & 1 & -1/2 \\ -1/2 & 1/2 & -1/2 & 1 \end{pmatrix} \text{ und } \begin{pmatrix} 1 & -1/2 & 1/2 & -1/2 \\ -1/2 & 1 & -1/2 & 0 \\ 1/2 & -1/2 & 1 & -1/2 \\ -1/2 & 0 & -1/2 & 1 \end{pmatrix}$$

auf. Es ist also für den (wählbaren) Startpunkt $\frac{1}{2}Q_{A_d}$, $d = 3, 4$, im Falle einer nicht vollständig positiven Matrix mit negativen Außendiagonalelementen, für eine Termination unerheblich, ob man Algorithmus 6 oder Algorithmus 7 betrachtet.

Weiter gibt uns diese Beobachtung, dass die Extremalstrahlen E_{ij} , $i \neq j$ zumindest für $d = 3, 4$ tatsächlich alle als Extremalstrahlen der Kegel $(\mathcal{V}(P))^*$ vorkommen. Man beachte, dass dies keine Folgerung von Theorem 4.1 ist.

Wir vermuten, dass sich dies auch für $d \geq 5$ wahr ist.

5 Das Kopositive Minimum

Wir wollen uns in diesem Kapitel mit der algorithmischen Berechnung des kopositiven Minimums einer strikt kopositiven Matrix Q sowie ihrer minimalen Vektoren beschäftigen. Im Folgenden bezeichnen wir dieses Problem mit *KoM*.

Im Fall positiv definiten Matrizen kann das zu KoM äquivalente Problem (Berechnung des arithmetischen Minimums einer positiv definiten Matrix Q und aller Vektoren, bei denen es angenommen wird, auch „shortest vector problem“ oder SVP genannt) durch einen Algorithmus von Fincke und Pohst [52, Algorithmus (2.12)] gelöst werden. Dieser berechnet für eine Konstante C alle ganzzahligen Vektoren x , für die $Q[x] \leq C$ ist. Indem man C immer dann reduziert, wenn ein solcher ganzzahliger Vektor gefunden wird, bestimmt man nach und nach das arithmetische Minimum und zusätzlich alle Vektoren, bei denen es angenommen wird. Es ist daher naheliegend, dass KoM mit ähnlichen Ideen und Ansätzen gelöst werden kann.

In [47, Sektion 3.5] und [112] wird je ein Algorithmus zum Lösen von KoM vorgestellt, die jeweils auf der Idee des Fincke-Pohst-Algorithmus basieren. Wir stellen im Folgenden zunächst den „klassischen“ Algorithmus aus [47, Sektion 3.5], welcher auf „Simplexpartitionierungen“ basiert, vor. Danach wollen wir uns mit dem, noch experimentellen, „neuen“ Algorithmus in [112] beschäftigen, der auf dem Lösen gewisser quadratischer Probleme beruht.

5.1 Klassische Berechnung des kopositiven Minimums

In dieser Sektion geben wir im Wesentlichen die Beschreibung in [47, Sektion 3.5] wieder.

Wie auch der R^d ist der nichtnegative Orthant $\mathbb{R}_{\geq 0}^d$ in einem gewissen Sinne „zu groß“, um ihn zu durchsuchen. Fincke und Pohst lösen dieses Problem in [54], indem sie die Cholesky-Zerlegung $Q = LL^T$ für positiv definite Matrizen verwenden [53, Sektion 4.3], um über „Backtracking“ obere Schranken für die Koeffizienten x_i der ganzzahligen Vektoren x , in Abhängigkeit der Konstante C , zu ermitteln. Eine solche Cholesky-Zerlegung existiert nur für positiv semidefinite Matrizen [53, Theorem 4.3.3]. Da es durchaus strikt kopositive Matrizen gibt, welche nicht positiv semidefinit sind (siehe auch Sektion 4.4), kann man dies im kopositiven Fall nicht verwenden.

Die Grundidee des Ansatzes in [47, Sektion 3.5] ist es, nun den „zu großen“ $\mathbb{R}_{\geq 0}^d$ in kleinere Kegel zu unterteilen und in jedem dieser kleineren Kegel einen modifizierten Fincke-Pohst-Algorithmus zu verwenden. Um diese Unterteilung zu konstruieren werden Simplexpartitionierungen verwendet, welche wir im Folgenden einführen werden.

5.1.1 Simplexpartitionierungen

Wie wir bereits in Sektion 3.2.3 angemerkt haben, liegt jeder nichtnegative Vektor auf einem Strahl, der von einem Vektor in Δ^S erzeugt wird. Dies ist einfach ersichtlich, denn bezeichne für einen nichtnegativen Vektor $x \neq 0$ $\mu = \sum_i x_i > 0$. Dann gilt für $y = \mu x \geq 0$, dass $\sum_i y_i = 1$ ist. Also ist $y \in \Delta^S$ und da $\mu > 0$ ist, liegt x auf dem von y erzeugten Strahl. Wir können uns also in einem gewissen Sinne auf den Standardsimplex Δ^S „zurückziehen“.

Indem wir dann eine Unterteilung von Δ^S bestimmen und die Kegel, welche durch diese Unterteilung erzeugt werden, betrachten, erhalten wir eine Unterteilung des $\mathbb{R}_{\geq 0}^d$ in „kleinere“ Kegel.

Dazu sei Δ ein Simplex im \mathbb{R}^d . Wir nennen eine Familie $P = \{\Delta_1, \dots, \Delta_n\}$ von Simplex eine *Simplexpartition* (oder kurz *Partition*) von Δ , wenn

$$\Delta = \bigcup_{i=1}^n \Delta_i \text{ und } \text{int}\Delta_i \cap \text{int}\Delta_j = \emptyset, \text{ für } i \neq j \text{ gilt.} \quad (5.1)$$

Wir bezeichnen weiter einen Vektor v als *Ecke der Partition* P , wenn v eine Ecke eines Simplexes Δ_i in P ist. Analog bezeichnen wir $[v, w]$ als *Kante der Partition*, wenn sie eine Kante einer der Simplexes ist.

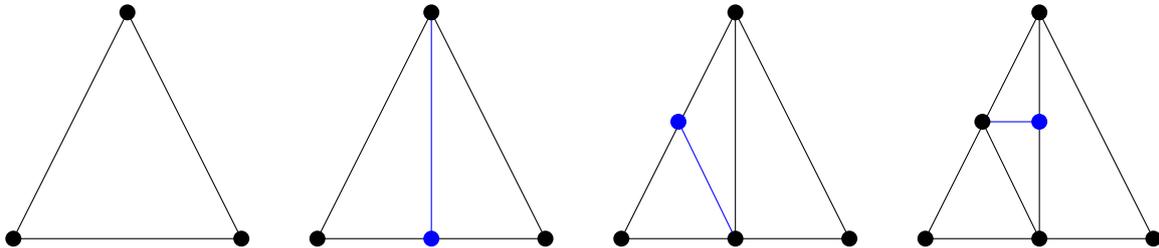


Abbildung 5.1: Sukzessive Partitionierung eines Simplexes

Im Folgenden beschreiben wir also, wie man eine für diese Aufgabe „geeignete“ Partition des Standardsimplexes bestimmt. Hat man schlussendlich eine Partition $P = \{\Delta_1, \dots, \Delta_n\}$ konstruiert, so erhält man durch $\{K_{\Delta_1}, \dots, K_{\Delta_n}\}$ eine Unterteilung von $\mathbb{R}_{\geq 0}^d$, wobei $K_{\Delta_i} = \text{cone}\Delta_i$ ist.

Geeignet heißt hierbei, dass für alle Ecken v_i, v_j eines Simplexes der Partition $v_i Q v_j > 0$ gelten soll. Man beachte, dass dies trotz Kopositivität, für den Standardsimplex nicht gelten brauch. Allerdings benötigen wir diese Eigenschaft in der folgenden Sektion.

In [29, Theorem 2] wurde gezeigt, dass eine solche Partition in dem hier relevanten Fall, dass Q strikt kopositiv ist, immer existiert. Bundfuss und Dür entwickelten in [29, Algorithmus 1 und 2] weiter einen Algorithmus zur Bestimmung einer solchen Partition. Wir folgen hier nun ihrer Beschreibung.

Die grundlegende Idee für ihren Algorithmus sind sogenannte „radiale Bisektionen“. Für einen Simplex $\Delta = \text{conv}\{v_1, \dots, v_d\}$ und ein $v = \sum_{i=1}^d \lambda_i v_i \in \Delta$, mit $\lambda_i \geq 0$ und $\sum_{i=1}^d \lambda_i =$

1, ist die Familie $P = \{\Delta_{i_1}, \dots, \Delta_{i_n}\}$ mit

$$\Delta_{i_j} = \text{conv} \{v_1, \dots, v_{i_j-1}, v, v_{i_j+1}, \dots, v_d\}, \text{ für alle } i_j \text{ mit } \lambda_{i_j} > 0,$$

eine Partition von Δ [66, Lemma 1], v heißt hierbei *Bisektionspunkt*. Falls v sich auf einer Kante $[v_i, v_j]$ befindet, wird $[v_i, v_j]$ *Bisektionskante* genannt.

Input : $Q \in \text{int } \mathcal{COP}^d$
Output: Simplexpartition $P = \{\Delta_1, \dots, \Delta_n\}$, sodass $(v_i^k)^T Q v_j^k \geq 0$ für alle $i, j = 1, \dots, d, k = 1, \dots, n$
 Setze $D = \{\Delta^S\}, P = \emptyset$
while $D \neq \emptyset$ **do**
 Wähle einen Simplex $\Delta = \text{conv}\{v_1, \dots, v_d\}$ aus D
 if $v_i^T Q v_j \geq 0$ für alle $i, j = 1, \dots, d$ **then**
 | Setze $\bar{D} = D \setminus \{\Delta\}, P = P \cup \{\Delta\}$
 else
 | Bestimme Bisektion Δ_1, Δ_2 von Δ
 | Setze $D = D \setminus \{\Delta\} \cup \{\Delta_1, \Delta_2\}$
 end
end
 Gib P aus

Algorithmus 8: Bestimmung einer geeigneten Simplexpartition P einer Matrix $Q \in \text{int } \mathcal{COP}^d$

Wir können Ecken v_i, v_j mit $v_i^T Q v_j$ als „negative Kante“ betrachten. Ziel ist es also, durch Bisektion solche negativen Kanten in „positive Kanten“ zu transformieren.

Dafür seien $a = Q[v_i] > 0, b = Q[v_j] > 0$ und $c = v_i^T Q v_j < 0$ für eine negative Kante $[v_i, v_j]$. In [29, Lemma 4] wurde gezeigt, dass der Bisektionspunkt $v = \lambda v_i + (1 - \lambda)v_j$ die gewünschte obige Eigenschaft hat, wenn λ aus dem Intervall $I = [\frac{c}{c-a}, \frac{b}{b-c}] \subset [0, 1]$ gewählt wird. Dann ist

$$v_i^T Q v = \lambda a + (1 - \lambda)c \geq \frac{c}{c-a}a + (1 - \frac{c}{c-a})c = 0.$$

Analog erhalten wir, dass $v^T Q v_j \geq 0$ ist. Wir erhalten also die zwei neuen positiven Kanten $[v_i, v]$ und $[v, v_j]$.

Man beachte, dass I nicht leer ist, wenn Q kopositiv ist. Andernfalls ist [29, Lemma 4 (b)]

$$Q[v_j - \frac{c}{a}v_i] = Q[v_j] - 2\frac{c}{a}v_i^T Q v_j + \frac{c^2}{a^2}Q[v_i] = b - 2\frac{c^2}{a} + \frac{c^2}{a} < 0.$$

In [29] wird die hier vorgestellte Methode für Kopositivitätstests verwendet. Daher motivieren die Autoren eine bestimmte Wahl des Bisektionspunktes beziehungsweise von λ . Da wir hier diese Partitionen für einen anderen Zweck verwenden, wollen wir

eine andere Wahl vorstellen, die für diese Anwendung besser geeignet ist.

Dafür ist zunächst zu sehen, dass $a, b > 0$ sind, da Q strikt kopositiv ist. Da weiterhin

$$Q[v_i + v_j] = a + 2c + b > 0 \text{ genau dann, wenn } c > \frac{1}{2}(-a - b) \text{ gilt,}$$

ist auch $c > \max\{-a, -b\}$ erfüllt. Daher ist $\frac{1}{2} \in I$, denn

$$\frac{1}{2} \geq \frac{c}{c-a} \Leftrightarrow c \geq -a \text{ und } \frac{1}{2} \leq \frac{b}{b-c} \Leftrightarrow -b \leq c.$$

Damit können wir $v = \frac{1}{2}(v_i + v_j)$ als Bisektionspunkt wählen.

Da wir die Ecken der Simplizes in der nächsten Sektion so skalieren wollen, dass sie nur ganzzahlige Einträge haben, ist dieser Bisektionspunkt hier besser geeignet als der in [29] gewählte. Das liegt daran, dass wir hier für die Skalierung auf Ganzzahligkeit nur die maximale Zweierpotenz in den Nennern ermitteln müssen, beziehungsweise sogar nur das Maximum der Nenner.

Algorithmisch lässt sich nun eine Bisektion $\Delta = \Delta_1 \cup \Delta_2$ durch Algorithmus 9 bestimmen. Man beachte, dass die Ecken der so in Algorithmus 9 konstruierten Simplizes

Input : Simplex $\Delta = \text{conv}\{v_1, \dots, v_d\}$ mit $v_i^T Q v_i \geq 0$ für alle i und $v_i^T Q v_j < 0$ für wenigstens ein i, j

Output: 'Q nicht kopositiv' oder Simplexpartitionierung $\Delta = \Delta_1 \cup \Delta_2$

Wähle eine negative Kante $[v_i, v_j]$ von Δ

Setze $v = \frac{1}{2}(v_i + v_j)$

Setze $\Delta_1 = \text{conv}\{v_1, \dots, v_{i-1}, v, v_{i+1}, \dots, v_d\}$

Setze $\Delta_2 = \text{conv}\{v_1, \dots, v_{j-1}, v, v_{j+1}, \dots, v_d\}$

Algorithmus 9: Bestimmung einer geeigneten Bisektion $\Delta = \Delta_1 \cup \Delta_2$, siehe [29, Algorithmus 2]

und somit die Ecken der in Algorithmus 8 konstruierten Simplexpartitionierung nur rationale Einträge haben.

In [29, Theorem 2] haben Bundfuss und Dür gezeigt, dass es für jede strikt kopositive Matrix Q ein $\varepsilon > 0$ gibt, sodass jede Partition $P = \{\Delta_1, \dots, \Delta_n\}$, deren *Durchmesser*

$$\delta(P) = \max_{i,j,k} \|v_i^k - v_j^k\|$$

höchstens ε ist, die von uns verlangte Nichtnegativitätseigenschaft hat. Dabei bezeichnet v_i^k eine Ecke des Simplizes Δ_k .

Damit lässt sich die Partition über sogenannte „Längste-Kante-Bisektionen“ zu konstruieren. Dabei wird immer die Kante $[v_i, v_j]$ in der Mitte, also $v = \frac{1}{2}(v_i + v_j)$, biseziert, für die $\|v_i - v_j\|$ über alle Kanten der Partition maximal ist.

Dass man dadurch tatsächlich eine Partition erhält, die den Anforderungen von [29, Theorem 2] genügt, ist ein Resultat von Kearfott. Dieser zeigte, dass der Durchmesser

einer so konstruierten Partition immer gegen 0 geht [74, Theorem 3.1]. Genauer zeigte er, dass der Durchmesser $\delta(P)$ nach k solchen Bisektionen

$$\delta(P) \leq \left(\frac{\sqrt{3}}{2}\right)^{\lfloor \frac{k}{n} \rfloor} \delta(\Delta)$$

ist. Hierbei ist n die Dimension des zu partitionierenden Simplex Δ und $\delta(\Delta)$ sein Durchmesser.

In unserem Fall hat Δ^S die Dimension $d-1$ und einen Durchmesser von $\sqrt{2}$, sodass wir nach k Iterationen eine Partition mit einem Durchmesser von höchstens $\left(\frac{\sqrt{3}}{2}\right)^{\lfloor \frac{k}{d-1} \rfloor} \sqrt{2}$ haben.

Da es nach [29, Theorem 2] ein festes ε gibt, müssen wir also höchstens so viele Schritte machen, dass $\varepsilon \geq \left(\frac{\sqrt{3}}{2}\right)^{\lfloor \frac{k}{d-1} \rfloor} \sqrt{2}$ ist. Man beachte, dass dieser Ansatz und der Ansatz in [29] recht verschieden sind und im Allgemeinen nicht zum gleichen Ergebnis führen.

Es ist aber, aufgrund von [29, Theorem 2] [74, Theorem 3.1], eine gute Idee in Algorithmus 8 alle k Iterationen eine Längste-Kante-Bisektion durchzuführen, da es in seltenen Fällen vorkommen kann, dass Algorithmus 8 eine Partition konstruiert, deren Durchmesser nicht gegen 0 geht. Dies können wir aber durch solche Bisektionen garantieren.

d	Durchschnittliche Zeit in s	Maximale Zeit in s	Größte Partition
5	0.006844	0.109375	47
6	0.03325	0.875	140
7	0.097239	5.71875	380
8	0.500782	97.75	1499
9	1.852739	370.265625	4564
10	5.561708	794.40625	11752

Tabelle 5.1: Aufwand von Algorithmus 8 für verschiedene Dimensionen. Getestet wurden hierbei jeweils 1500 $d \times d$ -Matrizen $P + N$, wobei P positiv definit und N nichtnegativ ist. N wurde so erzeugt, dass wir Nenner und Zähler gleichverteilt aus $\{0, \dots, 100\}$ beziehungsweise $\{1, \dots, 100\}$ erzeugt wurden und die Einträge dann zusätzlich durch das Maximum geteilt wurden. Für P haben wir auf ähnliche Weise eine Matrix A mit Zählern aus $\{-100, 100\}$ erzeugt, sodass $\text{rk}A = d$, und setzten dann $P = A^T A$

Wie man anhand von Tabelle 5.1 sieht, kann der Aufwand für die Bestimmung einer solchen Simplexpartition mit Algorithmus 8 durchaus groß werden. Es gibt allerdings mehrere Ansätze, die Möglichkeiten für eine Verbesserung bieten können. Auf diese wollen wir hier nicht mehr eingehen und verweisen stattdessen auf [41, Kapitel 9][122].

5.1.2 Hermitsche Normalform

Da wir nun eine Unterteilung $\{K_{\Delta_1}, \dots, K_{\Delta_n}\}$ des $\mathbb{R}_{\geq 0}^d$ haben, beschreiben wir in den folgenden Abschnitten, wie man einen Fincke-Pohst-artigen Algorithmus auf diese Kegel anwenden kann.

Zunächst lassen sich alle Elemente im Kegel K_{Δ} als positive Linearkombination der Ecken v_1, \dots, v_d des Simplexes Δ darstellen. Um also in diesem Kegel alle ganzzahligen Vektoren x mit $Q[x] \leq C$ zu finden, reicht es aus Koeffizienten $\alpha_1, \dots, \alpha_d \geq 0$ zu suchen, so dass $x = \sum_{i=1}^d \alpha_i v_i \in \mathbb{Z}^d$ und $Q[x] \leq C$ gilt.

Auf diese Weise können theoretisch, ähnlich wie beim Fincke-Pohst-Algorithmus, alle ganzzahligen Vektoren x im Kegel K_{Δ} mit $Q[x] \leq C$ bestimmt werden. Reduziert man dann wie beim Fincke-Pohst-Algorithmus C , wenn man solch ein x gefunden hat, kann das „Minimum“ von Q in diesem Kegel sowie alle „minimalen Vektoren“ in K_{Δ} bestimmt werden. Wiederholt man diesen Prozess für alle Kegel der Unterteilung, erhält man das koperative Minimum und die minimalen Vektoren der Matrix Q .

Das Problem besteht also aus dem Finden der Koeffizienten $\alpha_1, \dots, \alpha_d \geq 0$. Um dies zu ermöglichen, wird die Hermite-Normalform einer Matrix A verwendet.

Eine nicht-singuläre Matrix $B \in \mathbb{R}^{d \times d}$ ist in *Hermite-Normalform*, wenn B eine nichtnegative obere Dreiecksmatrix ist und der eindeutig bestimmte größte Eintrag jeder Spalte auf der Diagonalen ist [110, Kapitel 4.1].

Zwei der für uns wichtigen Eigenschaften der Hermite-Normalform sind zunächst, dass sie in unserem Fall eindeutig bestimmt ist, wie wir gleich sehen werden. Genauer gilt für eine nicht-singuläre ganzzahlige Matrix $A \in \mathbb{R}^{d \times d}$, dass es eine eindeutig bestimmte Matrix unimodulare $U \in \mathbf{GL}_d(\mathbb{Z})$ gibt, sodass $B = UA$ in Hermite-Normalform ist [110, Korollar 4.2a, 4.3b]. B heißt dann auch Hermite-Normalform von A .

Weiter lässt sich die Hermite-Normalform einer solchen Matrix in polynomieller Zeit [72, Theorem 2] bestimmen. Die generelle Strategie dabei basiert auf dem Gauß-Algorithmus. Man transformiert nacheinander durch elementare ganzzahlige Zeilenoperationen die Spalten der Matrix A in Hermite-Normalform. Das Produkt, der dabei entstehenden Elementarmatrizen ergibt dann die Matrix U .

Ein Problem, welches hierbei auftreten kann, ist die sogenannte „Zwischen-Eintrags-Explosion“ [60]. Dabei werden die Einträge in den noch nicht bearbeiteten Spalten durch die Zeilenoperationen der schon bearbeiteten sehr groß. Dies kann sich auf die Laufzeit auswirken, siehe auch [60].

In [60, Sektion 3] werden Ansätze diskutiert, um diesem Problem entgegen zu wirken. Der Ansatz, den wir benutzen wollen, beruht auf ggT-Berechnungen von n Zahlen. Bezeichnet a_i die i -te Spalte von A , dann bestimmt man im k -ten Schritt einen ganzzahligen Vektor x , sodass $x^T a_k = \text{ggT}(a_{kk}, \dots, a_{dk})$. Dann verwendet man dieses x und die k -te Zeile, um die restlichen Einträge auf 0 zu bekommen.

Damit die Einträge möglichst klein bleiben, will man einen möglichst kurzen Vektor x mit dieser Eigenschaft finden. Dies ist allerdings ein \mathcal{NP} -schweres Problem [60, Theorem 2]. In [61, Algorithmus 3] wird der LLL-Algorithmus [82] benutzt, um dieses Problem

(zumindest für 3 Zahlen) effizient zu lösen [61, Korollar 5.2]. Die dabei bestimmten Vektoren x müssen nicht notwendigerweise die Kürzesten sein, bilden aber eine „gute“ Approximation an diese, siehe [61][82].

Damit kann nun ein Algorithmus zur Bestimmung der Hermit-Normalform konstruiert werden, welcher die Zwischen-Eintrags-Explosion „verhindert“, siehe auch [61, Algorithmus 4]. Tatsächlich werden die Einträge während der gesamten Berechnung in [61, Algorithmus 4] höchstens so groß wie $2^{\mathcal{O}(d \log(dN))}$ [71, Theorem 1.1], wobei N das Quadrat der maximalen Norm der Zeilen von A ist. Für praktische Ergebnisse dieses Algorithmus verweisen wir auf [61, Sektion 7]

Wie bereits im vorherigen Abschnitt angemerkt sind die Ecken der zuvor bestimmten Simplexpartitionierung rational, sodass wir diese so skalieren können, dass sie ganzzahlig sind, siehe auch Kapitel 3. Man beachte, dass dies den Kegel K_Δ nicht verändert. Bezeichnen also v_1, \dots, v_d diese ganzzahligen Vektoren und V die Matrix mit den v_i als Spalten.

Dann ist V ganzzahlig und nicht singular, denn die Vektoren v_1, \dots, v_d spannen zusammen mit dem Nullvektor einen d -Simplex auf, sind also linear unabhängig. Damit existiert eine eindeutige Hermite-Normalform W von V und eine ganzzahlig unimodulare Matrix U , sodass $W = (w_1, \dots, w_d) = UV$ ist.

Wir betrachten nun statt Q die Matrix $Q' = U^{-\top}QU^{-1}$. Dann gilt zunächst, dass

$$0 \leq v_i^\top Q v_j = w_i^\top Q' w_j \quad (5.2)$$

ist, wobei diese Ungleichung offenbar strikt ist, wenn $i = j$ ist. Damit erhalten wir, dass

$$Q \left[\sum_{i=1}^d \alpha_i v_i \right] \leq M \text{ genau dann, wenn } Q' \left[\sum_{i=1}^d \alpha_i w_i \right] \leq M \text{ ist.}$$

Da die Matrix U ganzzahlig unimodular ist, also die induzierte Abbildung insbesondere ganzzahlige Vektoren wieder auf ganzzahlige Vektoren abbildet, ist

$$v = \sum_{i=1}^d \alpha_i v_i \text{ genau dann ganzzahlig, wenn } w = \sum_{i=1}^d \alpha_i w_i \text{ ganzzahlig ist.} \quad (5.3)$$

Für das Bestimmen der α_i macht es also „keinen“ Unterschied, ob man die v_i oder die w_i betrachtet. Da allerdings W in Hermite-Normalform ist, also insbesondere eine obere Dreiecksmatrix ist, vereinfacht sich der Ausdruck $\sum_i \alpha_i w_i$ zu

$$\left(\sum_{j=1}^d \alpha_j w_{j,1}, \sum_{j=2}^d \alpha_j w_{j,2}, \dots, \alpha_{d-1} w_{d-1,d-1} + \alpha_d w_{d,d-1}, \alpha_d w_{d,d} \right)^\top. \quad (5.4)$$

Dabei ist $w_{i,j}$ der j -te Eintrag von w_i .

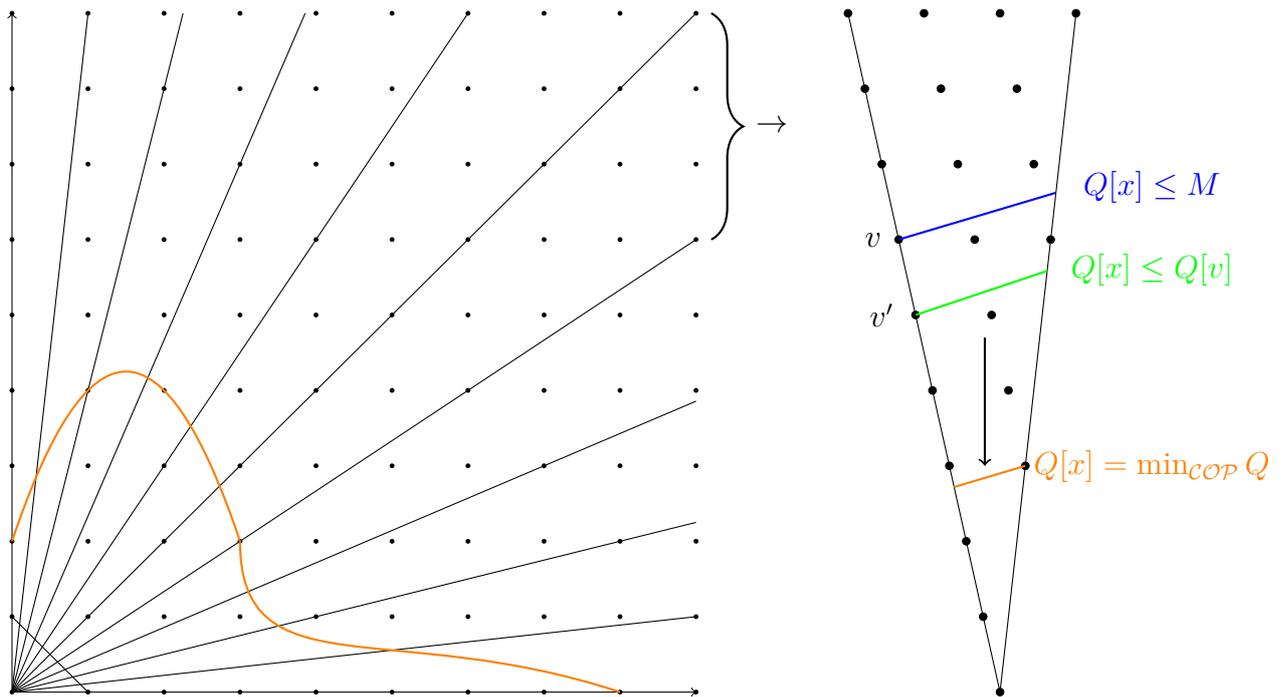


Abbildung 5.2: Bestimmung des kopositiven Minimums einer Matrix mittels Simplexpartitionierung

5.1.3 Berechnung der Koeffizienten α_i

Die Form von (5.4) erlaubt nun die α_i sukzessiv über eine Art „Backtracking“ zu bestimmen.

Für α_d gilt nach Voraussetzung, dass $\alpha_d \geq 0$ und $\alpha_d w_{d,d}$ ganzzahlig ist. Somit ist $\alpha_d \in \left\{ \frac{n}{w_{d,d}}, n = 1, 2, 3, \dots \right\}$. Weiter ist $\alpha_d \leq \left\lfloor \sqrt{\frac{C}{Q'[w_d]}} \right\rfloor$, denn es gilt

$$C \geq Q' \left[\sum_{i=1}^d \alpha_i w_i \right] \geq \alpha_d^2 Q'[w_d],$$

da $w_i^T Q' w_j \geq 0$ und $w_i^T Q' w_i > 0$ ist. Damit gibt es nur endlich viele Möglichkeiten für den Wert von α_d .

Fixieren wir einen Wert von α_d , lassen sich analog die möglichen Werte für $\alpha_k, k = d - 1, \dots, 1$ bestimmen. Genauer ergibt sich für α_k , dass

$$\alpha_k \in \left\{ \frac{n - \sum_{j=k+1}^d \alpha_j w_{j,k}}{w_{k,k}} \mid n = \left\lfloor \sum_{j=k+1}^d \alpha_j w_{j,k} \right\rfloor, \left\lfloor \sum_{j=k+1}^d \alpha_j w_{j,k} \right\rfloor + 1, \dots \right\} \text{ ist,} \quad (5.5)$$

wobei alle α_i mit $i > k$ bereits fixiert sind.

Input : Simplicialer Kegel K_Δ mit ganzzahligen „Ecken“ V , Matrix $Q' = U^{-\top}QU^{-1}$, Hermitesche Normalform W von V mit $UV = W$, Konstante $C > 0$

Output: $\min = \min_{v \in K_\Delta \cap \mathbb{Z}^d} Q[v]$ und $\text{MinV} = \{v \in K_\Delta \cap \mathbb{Z}^d \mid Q[v] = \min\}$

Setze $\text{MinV} = \emptyset$, $i = d$, $\alpha_k = 0$ für $k = 1, \dots, d$

Berechne $C_d = \sqrt{\min/Q[w_d]}$

Schritt 1: Falls $i > 1$ ist, gehe zu **Schritt 2**. Sonst gehe zu **Schritt 4**

Schritt 2: Falls $\alpha_i \leq C_i$ ist, setze $i = i - 1$, bestimme C_i mit (5.7), initialisiere α_i mit (5.5) und gehe zu **Schritt 1**. Sonst gehe zu **Schritt 3**

Schritt 3: Falls $i < d$ ist, setze $i = i + 1$ und erhöhe α_i wie in (5.5) und gehe zu **Schritt 1**. Sonst gibt C und MinV aus

Schritt 4: **while** $\alpha_1 \leq C_1$ **do**

Setze $w = \sum_i \alpha_i w_i$

if $Q'[w] = C$ und $w \neq 0$ **then**

Setze $v = \sum_i \alpha_i v_i$

Setze $\text{MinV} = \text{MinV} \cup \{v\}$

else if $Q'[w] < C$ und $w \neq 0$ **then**

Setze $v = \sum_i \alpha_i v_i$

Setze $C = Q'[w]$, $\text{MinV} = \{v\}$ und gehe zu **Schritt 1**

end

Erhöhe α_2 wie in (5.5), setze $i = 2$ und gehe zu **Schritt 2**

Algorithmus 10: Algorithmus zum Bestimmen des Minimums, sowie aller minimalen Vektoren in einem simplicialen Kegel

Weiter ist $\alpha_k \leq C_k$. Dabei wird C_k so bestimmt, dass

$$C \geq Q' \left[\sum_{j=k}^d \alpha_j w_j \right] = \sum_{j=k}^d \alpha_j^2 Q'[w_j] + 2 \sum_{j=k}^d \sum_{i=j+1}^d \alpha_j \alpha_i w_j^\top Q' w_i \text{ ist.} \quad (5.6)$$

Man beachte, dass durch die Fixierung von α_j für $j > k$, (5.6) eine quadratische Ungleichung in α_k mit den Parametern

$$a = Q'[w_k] > 0$$

$$b = 2 \sum_{i=k+1}^d \alpha_i w_k^\top Q' w_i \geq 0$$

$$c = \sum_{i=k+1}^d \alpha_i^2 Q'[w_i] + 2 \sum_{i=k+1}^d \sum_{j=i+1}^d \alpha_i \alpha_j w_i^\top Q' w_j \geq 0.$$

ist. Da wir die Bedingung $\alpha_k \geq 0$ haben, genügt es hier den Gleichheitsfall zu lösen und erhalten daher

$$C_k = \frac{\sqrt{4a(M - c) + b^2} - b}{2a}, \quad (5.7)$$

falls $M > c$ ist. Ansonsten setzen wir $C_k = 0$.

Damit lassen sich nun, wie in Algorithmus 10, alle möglichen Werte für $\alpha_1, \dots, \alpha_d$ und damit alle ganzzahligen Vektoren x im Kegel K_Δ mit $Q[x] \leq C$ bestimmen. Die Berechnung des kopositiven Minimums findet dann wie in Algorithmus 11 statt.

Man beachte, dass, wie in Tabelle 5.1, die Anzahl der zu durchsuchenden Kegel sehr groß werden kann. Da wir auf jeden dieser Kegel einen Fincke-Pohst-Algorithmus anwenden, ist ersichtlich, dass auch der Aufwand zum Lösen von KoM sehr groß werden kann. In [54, Sektion 3] geben die Autoren eine Komplexitätsanalyse für den Fincke-Pohst-Algorithmus an. Es wäre interessant zu sehen, ob sich der Aufwand des modifizierten Algorithmus (Algorithmus 10) in einer ähnlichen Größenklasse befindet.

Die initiale Festlegung von $\min C$ in Algorithmus 11 als das kleinste Diagonalelement von Q sichert, dass tatsächlich ganzzahlige Vektoren x mit $Q[x] \leq C$ existieren.

Andere Festlegungen sind hier durchaus denkbar und könnten Aufwand sparen. So ist beispielsweise die Festlegung $C = 1$ für alle in Algorithmus 1(6,7) vorkommenden kopositiven Matrizen durchaus möglich, da wir entweder schon wissen, dass das kopositive Minimum dieser Matrizen gleich 1 ist oder wir ein kopositives Minimum, welches kleiner als 1 ist, erwarten.

Die Menge S aus Algorithmus 4 kann ebenfalls mit einer Anpassung der Algorithmen 10 und 11 bestimmt werden.

```

Input :  $Q \in \text{int } \mathcal{COP}^d$ 
Output:  $\min C = \min_{\mathcal{COP}} Q$  und  $\text{MinC} = \text{Min}_{\mathcal{COP}} Q$ 
Bestimme eine Simplexpartitionierung  $P$  mit Algorithmus 8
Setze  $\min C = \min_{i=1, \dots, d} q_{ii}$  und  $\text{MinC} = \{e_i \mid q_{ii} = \min C\}$ 
for  $k = 1, \dots, n$  do
    Skaliere die Ecken von  $\Delta_k$ , sodass  $v_i^k \in \mathbb{Z}_{\geq 0}^d$  und setze  $V = (v_1^k, \dots, v_d^k)$ 
    Bestimme  $U \in \mathbf{GL}_d(\mathbb{Z})$ , sodass  $W = UV$  in hermitscher Normalform ist
    Setze  $Q' = U^{-\tau} Q U^{-1}$ 
    Bestimme die minimalen Vektoren  $\text{MinV}$  und das Minimum  $\min$  in  $\Delta_k$ 
    mit Algorithmus 10
    if  $\min < \min C$  then
        | Setze  $\min C = \min$ ,  $\text{MinC} = \text{MinV}$ 
    else if  $\min = \min C$  then
        | Setze  $\text{MinC} = \text{MinC} \cup \text{MinV}$ 
end
Gib  $\min C$  und  $\text{MinC}$  aus

```

Algorithmus 11: Bestimmung des kopositiven Minimums einer Matrix Q mittels Simplexpartitionierungen

Dies beendet schlussendlich auch unsere Analyse von Algorithmus 1.

5.2 Ein experimenteller Algorithmus

Wir wollen nun eine neuartige Methode von Schürmann [112] zum Lösen von KoM vorstellen, welche noch genauer untersucht werden muss, siehe Algorithmus 12.

Input : $Q \in \text{int } \mathcal{COP}^d$
Output: $\min_{\mathcal{COP}} Q$ und $\text{Min}_{\mathcal{COP}} Q$
 Setze $\min C = \min_{i=1, \dots, d} q_{ii}$, $\text{Min} C = \{e_i \mid q_{ii} = \min C\}$, $x = 0$, $i = 1$
Schritt 1: Bestimme
 $C_i = \max\{y_i \mid Q[y] \leq \min C, y_1 = x_1, \dots, y_{i-1} = x_{i-1}, y_i, \dots, y_d \in \mathbb{R}_{\geq 0}^d\}$
Schritt 2: Falls $i < d$ ist, gehe zu **Schritt 3**. Sonst gehe zu **Schritt 5**
Schritt 3: Falls $x_i \leq C_i$ ist, Setze $i = i + 1$, $x_i = 0$ und gehe zu **Schritt 1**.
 Sonst gehe zu **Schritt 4**
Schritt 4: Falls $i > 1$ ist, setze $i = i - 1$, $x_i = x_i + 1$ und gehe zu **Schritt 3**. Sonst gib $\min C$ und $\text{Min} C$ aus
Schritt 5: **for** $x_d = 0, \dots, \lfloor C_d \rfloor$ **do**
 if $Q[x] = \min C$ und $x \neq 0$ **then**
 | Setze $\text{Min} C = \text{Min} C \cup \{x\}$
 else if $Q[x] < \min C$ **then**
 | Setze $\min C = Q[x]$, $\text{Min} C = \{x\}$, $i = 1$ und gehe zu **Schritt 1**
end
 Setze $x_{d-1} = x_{d-1} + 1$, $i = d - 1$ und gehe zu **Schritt 2**

Algorithmus 12: Alternative Methode zur Berechnung des kopositiven Minimums einer strikt kopositiven Matrix Q , siehe [112]

Die Idee von Algorithmus 12 beruht ebenfalls auf dem Fincke-Pohst-Algorithmus. Anders als bei der klassischen Methode in Sektion 5.1 wird hier keine Unterteilung des $\mathbb{R}_{\geq 0}^d$ bestimmt, sondern es werden Schranken der Koeffizienten x_i durch Lösen quadratischer Probleme konstruiert.

In der folgenden Sektion wollen wir diese quadratischen Probleme betrachten und erste Überlegungen und Ansätze für das praktische Lösen dieser zusammentragen.

5.2.1 Erste Analyse des quadratischen Problems

Um eine Schranke für den Koeffizienten x_i in Algorithmus 12 zu erhalten, wird das quadratische Problem

$$\begin{aligned} & \min -e_i^T y, \\ & \text{so dass } Q[y] \leq C, \\ & (E_i \ 0) y = \begin{pmatrix} x_1 \\ \vdots \\ x_{i-1} \end{pmatrix} = x^i \\ & \text{und } y \geq 0 \end{aligned} \tag{5.8}$$

gelöst. Dabei ist (5.8) ein QCQP, welche wir bereits in Sektion 2.3.2. Zur Erinnerung geben wir hier nochmal die Form eines allgemeinen QCQP an.

$$\begin{aligned} & \min Q[x] + c^T x, \\ & \text{so dass } P_i[x] + a_i^T x \leq b_i, \quad i = 1, \dots, n \end{aligned} \quad (5.9)$$

Die Klasse der QCQPs ist sehr allgemein, wie wir bereits in Sektion 2.3.2 angemerkt haben. So lässt sich beispielsweise das Standard-Quadratische-Problem (2.11) als QCQP auffassen. Daher ist diese Problemklasse offenbar \mathcal{NP} -schwer. Tatsächlich lassen sich viele \mathcal{NP} -schwere Probleme als QCQPs darstellen, siehe auch [100, Kapitel 2]. Allerdings lassen sich auch sehr einfache Probleme, wie lineare Optimierungsprobleme, in die Form (5.9) bringen.

Für Problem (5.8) ist bisher noch nicht bekannt, ob es \mathcal{NP} -schwer ist. Dennoch ist zu vermuten, dass (5.8) recht schwer ist.

Bevor wir uns daher genauer mit Ideen zur Lösung von (5.8) beschäftigen, wollen wir kurz erste praktische Erfahrungen ohne Vorbereitungen vorstellen.

Wir haben ausgetestet, ob sich das Problem nicht schon recht gut mit standardmäßigen Lösern ausreichend approximieren lässt. Wir verwendeten dabei die *minimize*-Funktion des Python-Pakets *scipy.optimize*.

Zunächst erzeugten wir, wie in Tabelle 5.1, 4×4 -Matrizen $Q = P + N$, deren kopositives Minimum und minimalen Vektoren wir mit Algorithmus 11 bestimmten. Offenbar ist eine untere Schranke für den optimalen Wert C_1 von (5.8) in Algorithmus 12 der maximale Wert x_1 , unter den minimalen Vektoren der Matrix Q . Ähnliche untere Schranken erhalten wir für die Werte C_i mit $i > 1$.

Mit diesen Schranken prüften wir dann, wie sehr die berechneten Lösungen c_i von diesen Schranken abweichen, und wie viele Instanzen unterhalb und oberhalb der unteren Schranken liegen. Da die x_i in Algorithmus 12 ganzzahlig sind, ist es für Approximationen an den Wert C_i wichtig, dass diese wenigstens genauso groß wie C_i sind, da andernfalls mögliche minimale Vektoren „übergangen“ werden.

i	Durchschnittliche Abweichung $ c_i - u_i $	$ \{c_i < u_i\} $	$ \{c_i \geq u_i\} $
1	0.244	121	379
2	0.933151	471	29
3	0.715309	396	104
4	0.555759	263	237

Wie man sieht, ist ein „naiver“ Ansatz nicht unbedingt möglich, sodass wir das quadratische Problem (5.8) genauer analysieren müssen. Im Folgenden geben wir einige erste Ideen, mit denen eine möglicherweise gute Approximation an C_i erreicht werden kann.

Die „einfachste“ Instanz von (5.8) erhalten wir, wenn Q positiv semidefinit ist. In diesem Fall ist (5.8) ein konvexes Problem. Mit [85, Sektion 2.1] können wir dann (5.8) über ein SOCP (siehe auch Sektion 2.3.1) darstellen. Diese lassen sich, wie bereits in Sektion 2.3.1 angemerkt, sehr effizient und genau lösen, siehe auch [10][96].

Es bleibt also den Fall, dass Q nicht positiv semidefinit ist, zu betrachten. Dann ist (5.8) ein nicht konvexes QCQP. Für einen Überblick über nicht-konvexe QCQPs verweisen wir auf [84][100].

Wir wollen nun zwei erste Ansätze zum Lösen des nicht-konvexen Falles von (5.8) besprechen.

Man beachte zunächst, dass (5.8) nur eine nicht-konvexe Nebenbedingung $Q[x] \leq C$ hat. In [75, Algorithmus 1] wird eine approximative Lösungsmethode für solche QCQPs diskutiert. Dabei werden zunächst die semidefinite Relaxation

$$\begin{aligned} & \min -e_i^T y, \\ & \text{so dass } \langle Q, Y \rangle \leq C, \\ & \begin{pmatrix} E_i & 0 \end{pmatrix} y = x^i, \\ & y \geq 0, \\ & Y - yy^T \in \mathcal{S}_{\geq 0}^d \\ & \text{und } Y \in \mathcal{S}_{> 0}^d \end{aligned} \tag{5.10}$$

von (5.8) und das Problem

$$\begin{aligned} & \min Q[y] - C, \\ & \text{so dass } \begin{pmatrix} E_i & 0 \end{pmatrix} y = x^i \\ & \text{und } y \geq 0, \end{aligned} \tag{5.11}$$

gelöst, um jeweils eine untere und obere Schranke c_1, c_2 an den optimalen Wert C_i von (5.8) zu erhalten [75, Theorem 2, Lemma 4]. Dann wird durch wiederholtes Lösen des Problems

$$\begin{aligned} & \min Q[y] - C, \\ & \text{so, dass } -e_i^T y \leq \frac{c_1 + c_2}{2} \end{aligned} \tag{5.12}$$

der optimale Wert von (5.8) approximiert, indem wir $c_2 = \frac{c_1 + c_2}{2}$ setzen, wenn der optimale Wert von (5.12) nicht positiv ist. Ist der optimale Wert positiv, so setzen wir $c_1 = \frac{c_1 + c_2}{2}$. Dies wird solange wiederholt, bis der Abstand von c_1 und c_2 klein genug ist.

In [75, Theorem 17] wurde gezeigt, dass die so konstruierten Werte c_1, c_2 tatsächlich gegen den optimalen Wert C_i von (5.8) konvergieren.

Man beachte, dass (5.11) und (5.12) nicht-konvexe quadratische Probleme mit konvexen Nebenbedingungen sind. (5.10) hingegen ist ein (konvexes) konisches Optimierungsproblem über dem Kegel der positiv semidefiniten Matrizen. Für eine ausführliche Behandlung solcher „semidefiniten Probleme“ verweisen wir auf [81]. Es gibt also eine gut untersuchte Lösungstheorie für solche Probleme.

Damit bleiben die Probleme (5.11) und (5.12). Die Autoren von [75] verwenden zum Lösen dieser Probleme einen in [38] entwickelten adaptiven Algorithmus (siehe auch [75, Algorithmus 2]), auf den wir hier nicht weiter eingehen wollen.

Es eröffnet sich hier die Frage, inwieweit die strikte Kopositivität von Q dazu beitragen kann, die Probleme (5.11) und (5.12) beispielsweise über die obige Methode zu

lösen.

Der zweite Ansatz, den wir hier ansprechen wollen, folgt mit den Techniken aus Sektion 2.3.2. Zunächst ist der zulässige Bereich von (5.8)

$$Z_i = \{y \in \mathbb{R}^d \mid Q[y] \leq C, A_i y = \bar{x}^i, y \geq 0\}.$$

Viele Lösungsansätze für QCQPs lassen sich dahingehend interpretieren, eine berechenbare Darstellung der Menge $Z_i^* = \text{clconv} \{(y, Y) \mid y \in Z_i, Y = yy^T\}$ zu finden [34].

Mit den Techniken aus Sektion 2.3.2 erhalten wir zumindest, dass

$$Z_i^* \subseteq \left\{ (y, Y) \mid \begin{pmatrix} Y & y \\ y^T & 1 \end{pmatrix} \in \mathcal{CP}^{d+1}, \langle Q, Y \rangle \leq C, (E_i \ 0) y = x^i, \right. \\ \left. \langle Y, E_{jj} \rangle = (x_j^i)^2, j = 1, \dots, i-1 \right\} = \mathcal{Z}_i \tag{5.13}$$

ist. \mathcal{Z}_i selber ist offenbar konvex und abgeschlossen. Es wäre interessant zu sehen, ob hier auch Gleichheit gilt oder wir mit \mathcal{Z}_i bereits eine gute Approximation an C_i berechnen können. Mit (5.13) gilt sicherlich für den optimalen Wert c_i des Problems über \mathcal{Z}_i , dass $C_i \leq c_i$ ist. Wie wir bereits bei den vorherigen Tests anmerkten, ist diese Eigenschaft für approximative Lösungen von (5.8), aufgrund der Ganzzahligkeit der x_i , eine „Mindestvoraussetzung“.

Da die Zielfunktion in (5.8) linear ist, ergibt sich hier ein vollständig positives Problem. Wie wir in Sektion 2.3.3 gesehen haben, können wir solche Probleme, unter gewissen Voraussetzungen, durch lineare Probleme approximieren.

Bei diesem Ansatz stellt sich natürlich die Frage, ob dies den Lösungsprozess von (5.8) nicht unnötig verkompliziert.

Offenbar sind beide hier vorgestellten Ansätze sehr grobe erste Ideen, welche sehr viel genauer untersucht werden müssten, als es der Rahmen dieser Arbeit erlauben würde. Weiter gibt es eine Vielzahl von Ansätzen zum Lösen nicht-konvexer QCQPs, auf die wir hier ebenfalls nicht eingehen konnten. Dafür verweisen wir auf [34][84][75][100] und den Referenzen darin.

6 Fazit und Ausblick

Wir haben uns in dieser Arbeit mit grundlegenden Konzepten und Objekten aus der konvexen und diskreten Geometrie sowie der konischen Optimierung vertraut gemacht. Die Konzepte haben wir besonders in Bezug auf die Kegel der kopositiven und vollständig positiven Matrizen angewandt und dabei auf einige offene Probleme aufmerksam gemacht.

Ebenso haben wir uns mit klassischen und neuen Algorithmen und Problemen der computergestützten Geometrie beschäftigt, mit denen dann Algorithmus 1 und 6 sowie alle benötigten Komponenten verständlich und detailliert dargestellt haben.

Unsere Beiträge finden sich größtenteils in Kapitel 4, wo wir Algorithmus 6 mittels Theorem 4.1 verbessert haben und zeigen konnten, dass dieser das \mathcal{CP} -Membership-Problem für $d = 2$ löst. Auch haben wir auf mögliche Schwierigkeiten und Richtungen für einen Beweis von Vermutung 4.2 aufmerksam gemacht. Weiterhin haben wir in Sektion 5.2 erste Ideen für die Lösung des dort vorkommenden quadratischen Problems entwickelt.

Zusätzlich haben wir in dieser Arbeit eine große Menge an Literatur gesammelt, die sich für weiterführendes Lesen in die verschiedenen, in dieser Arbeit vorkommenden, Themenbereiche eignet.

Des Weiteren haben wir weitestgehend alle für Algorithmus 1 und Algorithmus 6 benötigten Algorithmen einschließlich dieser in Python implementiert [36]. Ein Projekt für die Zukunft ist, diese Implementationen zu optimieren. Bisher ist uns nur eine Implementation einer (grundlegenden) Version der double-description Methode gelungen, sodass wir noch eine (grundlegende) Version des reverse-search-Algorithmus implementieren wollen.

Im Verlauf dieser Arbeit haben wir auf mehrere offene Probleme und Fragen in den verschiedenen Bereichen dieser Arbeit aufmerksam gemacht, die wir für mögliche zukünftige Projekte halten. Diese führen wir daher hier noch einmal zusammenfassend auf.

- Möglichkeit der Umformulierung allgemeiner quadratischer Probleme als vollständig positive und kopositive Probleme, siehe Sektion 2.3.2
- Klassifizierung der Extremalstrahlen des kopositiven Kegels \mathcal{COP}^d für $d \geq 7$, siehe Sektion 2.4
- Beweis von [115, Vermutung 1], siehe Kapitel 3. Auch gibt es viele offene Probleme bezüglich des \mathcal{CP} -Rangs, siehe [113][115]
- Möglichkeit der Konstruktion einer möglichst guten Startmatrix für Algorithmus 1, siehe Sektion 3.2

- Konstruktion eines Algorithmus zum Bestimmen aller Extremalstrahlen eines punktierten polyedrischen Kegels, welcher polynomiell in Ein- und Ausgabegröße ist, siehe Sektion 3.2.1. Damit zusammenhängend die Konstruktion einer Auswahlregel der double-description Methode, sodass sie zu einem solchen Algorithmus wird, siehe Sektion 3.2.1
- Konstruktion einer „guten“ Pivotregel für Algorithmus 1, siehe Sektion 3.2.1
- Konstruktion einer kopositiven Version von [119, Lemma 9], siehe Sektion 3.2.2
- Konstruktion eines Kopositivitätstest für Matrizen der Form $P + R$, siehe Sektion 3.2.3
- Beweis von Vermutung 4.2 für $d \geq 3$, siehe Sektion 4.2
- Untersuchung von Vermutung 4.5 und Vermutung 4.6, siehe Sektion 4.4
- Optimierung von Algorithmus 8, siehe Sektion 5.1.1
- Untersuchung der Komplexität von Algorithmus 10, siehe Sektion 5.1.3
- Untersuchung des Problems (5.8) und Konstruktion einer geeigneten Lösungsmethode, siehe Sektion 5.2.1

Literaturverzeichnis

- [1] A. AFONIN, R. HILDEBRAND UND P.J.C. DICKINSON. *The extreme rays of the 6×6 copositive cone*. J. Glob Optim., <https://doi.org/10.1007/s10898-020-00930-y>, 2020.
- [2] K.M. ANSTREICHER. *Testing copositivity via mixed-integer linear programming*. Linear Algebra Appl. **609**, p. 218-230, 2021.
- [3] D. AVIS. *lrs: A revised Implementation of the reverse search vertex enumeration algorithm*. 1996.
- [4] D. AVIS. <http://cgm.cs.mcgill.ca/~avis/C/lrs.html>
- [5] D. AVIS, D. BREMNER UND R. SEIDEL. *How good are convex hull algorithms?*. Comput. Geometry **7** p. 265-301, 1997.
- [6] R. BADENBROEK UND E. DE KLERK. *An Analytic Center Cutting Plane Method to Determine Complete Positivity of a Matrix*. arXiv **2006.05319**, 2020.
- [7] R. BAGNARA, P.M. HILL UND E. ZAFFANELLA. *The Parma Polyhedral Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems*. Science of Comp. Prog. **72** p. 3-21, 2008.
- [8] A. BARVINOK. *A Course in Convexity*. Graduate Studies in Mathematics **54** American Mathematical Society, Providence, RI, 2002.
- [9] L.D. BAUMERT. *Extreme copositive quadratic forms*. Pac. Jour. Math. **19** p. 197-204, 1966.
- [10] A. BEN-TAL UND A NEMIROVSKI. *Lectures on Modern Convex Optimization*. SIAM, 2001.
- [11] A.M. BERGÉ, J. MARTINET UND F. SIGRIST. *Une généralisation de l'algorithme de Voronoï pour les formes quadratiques*. Journées arithmétiques de Genève, Astérisque **209** p. 137-158, 1992.
- [12] A. BERMAN, M. DÜR UND N. SHAKED-MONDERER. *Open problems in the theory of completely and copositive matrices*. Elect. Jour. Linear Algebra **29** p. 46-58, 2015.
- [13] A. BERMAN UND U.G. ROTHBLUM. *A note on the computation of the CP-Rank*. Linear Algebra Appl. **419** p. 1-7, 2006.

- [14] A. BERMAN UND N. SHAKED-MONDERER. *Completely Positive Matrices*. World Scientific Publishing , Singapore, 2003.
- [15] A. BERMAN UND N. SHAKED-MONDERER. *Completely Positive Matrices: Real, Rational, and Integral*. Acta Mathematica Vietnamica, 2018.
- [16] W.A. BLANKINSHIP. *A New Version of the Euclidean Algorithm*. Amer. Math. Mon. **70** p. 742-745, 1963.
- [17] I.M. BOMZE. *Linear-time copositivity detection for tridiagonal matrices and extension to block-tridiagonality*. SIAM J. Matrix Anal. Appl. **21** p. 840-848, 2000.
- [18] I.M. BOMZE. *Copositive optimization - recent developments and applications*. Eur. Jour. Oper. Research **216** p. 509-520, 2012.
- [19] I.M. BOMZE UND E. DE KLERK. *Solving Standard Quadratic Optimization Problems via Linear, Semidefinite and Copositive Programming*. J. Global Optim. **24** p. 163-185, 2002.
- [20] I.M. BOMZE, M. DÜR, E. DE KLERK, C. ROOS, A. QUIST UND T. TERLAKY. *On Copositive Programming and Standard Quadratic Optimization Problems*. J. Global Optim. **18** p. 301-320, 2000.
- [21] I.M. BOMZE, M. DÜR UND C.P. TEO. *Copositive Optimization*. OPTIMA **89** p. 2-8, 2012.
- [22] I.M. BOMZE, G. EICHFELDER. *Copositivity detection by difference-of-convex decomposition and ω -subdivision*. Math. Prog. **138** p. 365-400, 2013.
- [23] I.M BOMZE, F. JARRE UND F. RENDL. *Quadratic factorization heuristics for copositive programming*. Math. Prog. Comp. **3** p. 37-57, 2011.
- [24] I.M. BOMZE UND W. SCHACHINGER. *Multi-Standard Quadratic Optimization: interior point methods and cone programming reformulation*. Comput. Optim. Appl. **45** p. 237-256, 2010.
- [25] I.M. BOMZE, W. SCHACHINGER UND G. UCHIDA. *Think co(mpletely)positive! Matrix properties, examples and a clustered bibliography on copositive Optimization*. J. Global Optim. **52** p 423-445, 2012.
- [26] I.M. BOMZE, W. SCHACHINGER UND R. ULLRICH. *From seven to eleven: Completely positive matrices with high cp-rank*. Linear Algebra Appl. **459** p. 208-221, 2014.
- [27] S. BOSE UND E. SLUD. *Maximin efficiency-robust tests and some extensions*. Jour. Stat. Planning and Inference **46** p. 105-121, 1995.
- [28] A. BRÜNGGER, A. MARZETTA, K. FUKUDA UND J. NIEVERGELT. *The parallel search bench ZRAM and its applications*. Ann. Oper. Research **90** p. 45-63, 1999.

- [29] S. BUNDFUSS UND M. DÜR. *Algorithmic copositivity detection by simplicial partition*. Linear Algebra Appl. **428** p. 1511-1523, 2008.
- [30] S. BUNDFUSS UND M. DÜR. *An adaptive linear approximation algorithm for copositive programs*. SIAM Jour. Optim. **20** p. 30-53, 2009.
- [31] S. BURER. *On the copositive representation of binary and continuous nonconvex quadratic programs*. Math. Program. Ser. A **120** p. 479-495, 2009.
- [32] S. BURER. *Copositive Programming*. International Series Oper. Research Man. Science p. 201-218, Springer, 2011.
- [33] S. BURER, K.M. ANSTREICHER UND M. DÜR. *The difference between 5×5 double nonnegative and completely positive matrices*. Linear Algebra Appl. **431** p. 1539-1552, 2009.
- [34] S. BURER UND H. DONG. *Representing quadratically constrained quadratic programs as generalized copositive programs*. Oper. Res. Letters **40** p. 203-206, 2012.
- [35] R.W. COTTLE, G.J. HABETLER UND C.E. LEMKE. *On classes of copositive matrices*. Linear Algebra Appl. **3** p. 295-310, 1970.
- [36] V.F. DANNENBERG. *CopositiveProgramming*. <https://github.com/VFDannenberg/CopositiviteProgramming>, 2020.
- [37] E. DE KLERK UND D.V. PASECHNIK. *Approximation of the stability number of a graph via copositive programming*. SIAM Jour. Optim. **12** p. 875-892, 2002.
- [38] Z. DENG, S.C. FANG, Q. JIN UND C. LU. *Conic approximation to nonconvex quadratic programming with convex quadratic constraints*. J. Global Optim. **61**, p. 459-478, 2015.
- [39] P.H. DIANANDA. *On non-negativ forms in real variable some or all of which are non-negative*. Math. Proc. Cambridge Phil. Soc. **58** p. 17-25, 1962.
- [40] P.J.C. DICKINSON. *Geometry of the copositive and completely positive cone*. J. Math. Anal. Appl **380** p. 377-395, 2011.
- [41] P.J.C. DICKINSON. *The copositive cone, the completely positive cone and their generalisations*. Doktorarbeit, Universität Groningen, 2013.
- [42] P.J.C. DICKINSON UND M. DÜR. *Linear-time complete positivity detection and decomposition of sparse matrices*. SIAM J. Matrix Anal. Appl. **33** p. 701-720, 2012.
- [43] P.J.C. DICKINSON UND L. GIJZEN. *On the computational complexity of membership problems for the completely positive cone and its dual*. Comput. Optim. Appl. **57** p. 403-415, 2014.
- [44] J.H. DREW, C.R. JOHNSON UND R. LOEWY. *Completely Positive Matrices Associated with M-Matrices* Linear Multilinear Algebra **37** p. 303-310, 1994.

- [45] I. DUKANOVIC UND F. RENDL. *Copositive programming motivated bounds on the stability and the chromatic numbers*. Math. Program. Ser. A **121** p. 249-268, 2010.
- [46] M. DUTOUR SIKIRIĆ, A. SCHÜRMAN UND F. VALLENTIN. *Rational factorizations of completely positive matrices*. Linear Algebra Appl. **523** p. 46-51, 2017.
- [47] M. DUTOUR SIKIRIĆ, A. SCHÜRMAN UND F. VALLENTIN. *A simplex algorithm for rational CP-Factorization*. Math. Program., <https://doi.org/10.1007/s10107-020-01467-4>, 2020.
- [48] M. DÜR. *Copositive Programming - a Survey*. Recent Advances in Optimization and its Applications in Engineering p. 3-20, Springer, 2010.
- [49] M. DÜR UND G. STILL. *Interior points of the completely positive cone*. Elect. Jour. Linear Algebra **17** p. 48-53, 2008.
- [50] G. EICHFELDER UND J. JAHN. *Set-Semidefinite Optimization*. J. Convex. Anal. **15** p. 767-801, 2008.
- [51] V. ELSER. *Matrix product constraints by projection methods*. J. Global Optim. **68** p.329-355, 2017
- [52] U. FINCKE UND M. POHST. *Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis*. Math. Comp. **44** p. 463-471, 1985.
- [53] R.W. FREUND UND R.H.W. HOPPE. *Stoer/Bulirsch: Numerische Mathematik 1*. Springer, 2007.
- [54] K. FUKUDA UND A. PRODON. *Double Description Method Revisited*. Combinatorics and Computerscience p. 91-111, 1995.
- [55] J.W. GADDUM. *Linear inequalities and quadratic forms*. Pacific J. Math. **8** p. 411 - 414, 1958.
- [56] P. GROETZNER UND M. DÜR. *A factorization method for completely positive matrices*. Linear Algebra Appl. **591** p. 1-24, 2020.
- [57] N. GVOZDENOVIĆ UND M. LAURENT. *The operator Ψ for the chromatic number of a graph*. SIAM Jour. Optim. **19** p. 572-591, 2008.
- [58] M. HALL JR.. *Combinatorial Theory*. Wiley, 1998.
- [59] A. HATCHER. *Topology of Numbers*. Buch in Vorbereitung, <http://pi.math.cornell.edu/~hatcher/TN/TNpage.html>, 2020.
- [60] G. HAVAS UND B.S. MAJEWSKI. *Hermite normal form computation for integer matrices*. Congr. Numer. **105** p. 87-96, 1994.

-
- [61] G. HAVAS, B.S. MAJEWSKI UND K.R. MATTHEWS. *Extended GCD and Hermite Normal Form Algorithms via Lattice Basis Reduction*. *Exper. Math.* **7** p. 125-136, 1998.
- [62] E. HAYNSWORTH UND A.J. HOFFMAN. *Two Remarks on Copositive Matrices*. *Linear Algebra Appl.* **2** p. 387-392, 1969.
- [63] R. HILDEBRAND. *The extreme rays of the 5×5 copositive cone*. *Linear Algebra Appl.* **437** p. 1538-1547, 2012.
- [64] R. HILDEBRAND. *Minimal zeros of copositive matrices*. *Linear Algebra Appl.* **437** p. 1538-1547, 2014.
- [65] J.-B. HIRIART-URRUTY UND A. SEEGER. *A variational approach to copositive matrices*. *SIAM Review* **52** p. 593-629, 2010.
- [66] R. HORST. *A new branch and bound approach for concave minimization problems*. *Lecture Notes in Computer Science* **41** p. 330-337, Springer, 1975.
- [67] A. HURWITZ. *Über die Reduktion der binären quadratischen Formen*. *Math. Ann.* **45** p. 85-117, 1894.
- [68] K. IKRAMOV. *Linear-time algorithm for verifying the copositivity of an acyclic matrix*. *Comp. math. mathematical physics* **42** p. 1701-1703, 2002.
- [69] F. JARRE UND K. SCHMALLOWSKY. *On the computation of C^* certificates*. *J. Global Optim.* **45** p. 281-296, 2009.
- [70] C.R. JOHNSON UND R. REAMS. *Constructiong copositive matrices from interior matrices*. *Elect. Jour. Linear Algebra* **17** p. 9-20, 2008.
- [71] W. VAN DER KALLEN. *Complexity of the Havas, Majewski, Matthews LLL Hermite Normal Form Algorithm*. *J. Symb. Comp.* **30** p. 329-337, 2000.
- [72] R. KANNAN UND A. BACHEM. *Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix*. *SIAM J. Comput.* **8** p. 499-507, 1979.
- [73] R.M. KARP. *Reducibility among combinatorial problems*. *Complexity of Computer Computations* p. 85-103, R.E. Miller, J.W. Thatcher J.D. Bohlinger (eds.), New York, 1972.
- [74] B. KEARFOTT. *A Proof of Convergence and an Error Bound for the Method of Bisection in \mathbb{R}^n* . *Math. Comp.* **32** p. 1147-1153, 1978.
- [75] M. KEYANPOUR UND N. OSMANPOUR. *On solving quadratically constrained quadratic programming problem with one non-convex constraint*. *OPSEARCH* **55**, p. 320-336, 2018.

- [76] C. KELLY. *A Test of the Markovian Model of DNA Evolution*. Biometric **50** p. 653-664, 1994.
- [77] N. KOGAN UND A. BERMAN. *Characterization of completely positive graphs*. Disc. Math. **114** p.297-304, 1993.
- [78] T. LAFFEY UND H. ŠMIGOC. *Integer completely positive matrices of order two*. arXiv **1802.04129**, 2018.
- [79] J.L. LAGRANGE. *Démonstration d'un théorème d'arithmétique*. Oeuvres de Lagrange III p. 189-201, Nouv. Mém. Ann. Berlin, 1770.
- [80] J.B. LASSERRE. *Global optimization with polynomials and the problem of moments*. SIAM Jour. Optim. **11** p. 796-817, 2000.
- [81] M. LAURENT UND F. VALLENTIN. *Semidefinite Optimization*. Buch in Vorbereitung, 2016.
- [82] A.K. LENSTRA, H.W. LENSTRA JR. UND L. LOVÁSZ. *Factoring Polynomials with Rational Coefficients*. Math. Ann. **261** p. 515-534, 1982.
- [83] P. LI UND Y.Y. FENG. *Criteria for copositive matrices of order four*. Linear Algebra Appl. **194** p. 109-124, 1993.
- [84] J. LINDENROTH. *A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs*. Math. Programm. Ser. B **103**, p. 251-282, 2005.
- [85] M.S. LOBO, L. VANDENBERGHE, S. BOYD UND H. LEBRET. *Applications of second order cone programming*. Linear Algebra Appl. **284** p. 193-228, 1998.
- [86] P.N. MALKIN. *Computing Markov bases, Gröbner bases, and extrem rays*. Doktorarbeit, Université catholique de Louvain, 2007.
- [87] O. MASON UND R. SHORTEN. *On Linear Copositive Lyapunov Functions and the Stability of Switched Positive Linear Systems*. IEEE Transactions on Automatic Control **52** p. 1346-1349, 2007.
- [88] P. McMULLEN. *The maximum numbers of faces of a convex polytope*. Jour. Pure and Applied Math. **17** p. 179-184, 1970.
- [89] W.B. McRAE UND E.R. DAVIDSON. *An algorithm for the extreme rays of a pointed convex polyhedral cone*. SIAM J. Comput. **2** p. 281-293, 1973.
- [90] D.A. MILLER UND S.W. ZUCKER. *Copositive-plus Lemke algorithm solves polymatrix games*. Oper. Res. Letters **10** p. 285-290, 1991.
- [91] NATIONAL BUREAU OF STANDARDS. *Report 1818, Quarterly Report*. April - Juni 1952.

-
- [92] T.S. MOTZKIN, H. RAIFFA, G.L. THOMPSON UND R. THRALL. *The double description method*. Contributions to the Theory of Games **2** p. 51-74, 1953.
- [93] T.S. MOTZKIN UND E.G. STRAUS. *Maxima for graphs and a new proof of a theorem of Turán*. Canadian Journal of Mathematics **17** p. 533-540, 1965.
- [94] K.G. MURTY UND S.N. KABADI. *Some \mathcal{NP} -complete problems in quadratic and nonlinear programming*. Math. Program. **39** p. 117-129, 1987.
- [95] K. NATARAJAN, C.P. TEO UND Z. ZENG. *Mixed zero-one linear programs under objective uncertainty: a completely positive representation*. Oper. Res. **59** p. 713-728, 2011.
- [96] Y.E. NESTEROV UND A. NEMIROVSKI. *Interior-point polynomial methods in convex programming*. Studies in Appl. Math. **13**, SIAM, Philadelphia, 1994.
- [97] J. NIE. *The \mathcal{A} -truncated K -moment problem*. Found. Comput. Math. **14** p. 1243-1276, 2014.
- [98] J. NIE, Z. YANG UND X. ZHANG. *A complete semidefinite algorithm for detecting copositive matrices and tensors*. SIAM J. Optim. **28** p. 2902-2921, 2018.
- [99] J. OPGENORTH. *Dual cones an the Voronoi Algorithm*. Experimental Math. **10** p. 599-608, 2001.
- [100] J. PARK UND S. BOYD. *General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming*
- [101] P.A. PARRILO. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. Doktorarbeit, California Institute of Technology, 2000.
- [102] J. PEÑA, J. VERA UND L.F ZULUAGA. *Computing the stability number of a graph via linear and semidefinite programming*. SIAM Jour. Optim. **18** p. 87-105, 2007.
- [103] J. POVH UND F. RENDL. *A copositive programming approach to graph partitioning*. SIAM Journal on Optimization **18** p. 223-241, 2007.
- [104] J. POVH UND F. RENDL. *Copositive and semidefinite relaxations of the Quadratic Assignment Problem*. Discrete Optim. **6** p. 231-241, 2009.
- [105] J.C. PREISIG. *Copositivity and the minimization of quadratic functions with non-negativity and quadratic equality constraints*. SIAM Jour. Contr. Optim. **34** p. 1135-1150, 1996.
- [106] A.J. QUIST, E. DE KLERK, C. ROOS UND T. TERLAKY. *Copositive relaxations for general quadratic programming*. Optimazation Methods and Software **9** p. 185-208, 1998.

- [107] M.O. SAYIN UND T. BAŞAR. *Optimal Hierarchical Signaling for Quadratic Cost Measures and General Distributions: A Copositive Program Characterization*. arXiv **1907.09070**, 2019.
- [108] R. SEIDEL. *A convex hull algorithm optimal for point sets in even dimensions*. Doktorarbeit, University of British Columbia, 1980.
- [109] H.C. SIMPSON UND S.J. SPECTOR. *On copositive matrices and strong ellipticity for isotropic materials*. Arch. Rat. Mech. Anal. **84** p. 55-68, 1983.
- [110] A. SCHRIJVER. *Theory of Linear and Integer Programming*. Wiley, 1986.
- [111] A. SCHÜRMAN. *Computational geometry of positive definite quadratic forms*. American Mathematical Society, Providence, RI, 2009.
- [112] A. SCHÜRMAN. *Computing the copositive minimum*. In Bearbeitung.
- [113] N. SHAKED-MONDERER. *On the DJL conjecture for order 6*. Operators and Matrices **11**, p. 71-88, 2017.
- [114] N. SHAKED-MONDERER, I.M. BOMZE, F. JARRE UND W. SCHACHINGER. *On the cp-rank and minimal cp factorizations of a completely positive matrix*. SIAM Jour. Matrix Analysis Appl. **34** p. 355-368, 2013.
- [115] N. SHAKED-MONDERER, M. DÜR UND A. BERMAN. *Complete positivity over the rationals*. Pure and Applied Functional Analysis **3** p. 681-691, 2018.
- [116] Y. SHITOV. *Linear mappings preserving the copositive cone*. arXiv **1911.10553**, 2019.
- [117] J. SPONSEL UND M. DÜR. *Factorization and cutting planes for completely positive matrices by copositive projection*. Math. Program. Ser. A **143** p. 211-229, 2014.
- [118] M. TERZER UND J. STELLING. *Accelerating the Computation of Elementary Modes Using Pattern Trees*. Alg. in Bioinformatics, WABI p. 333-343, 2006.
- [119] W.P.J. VAN WOERDEN. *Perfect Quadratic forms: an Upper Bound and Challenges in Enumeration*. Masterarbeit, Leiden University, 2018.
- [120] W. XIA, J.C. VERA UND L.F. ZULUAGA. *Globally solving non-convex quadratic programs via linear integer programming techniques*. INFORMS Jour. Comp., 2018.
- [121] E.A. YILDIRIM. *On the accuracy of uniform polyhedral approximations of the copositive cone*. Optim. Meth. Software **27** p. 155-173, 2012.
- [122] J. ŽILINSKAS UND M. DÜR. *Depth-first simplicial partition for copositivity detection, with an application to MaxClique*. Optim. Meth. Software **26** p. 499-510, 2011.

Index

- Ecke, 9
- Facette, 9
- Halbraum, 3
- Hermite-Normalform, 70
- Hyperebene, 3
- Hülle, 4
 - konische, 4
 - konvexe, 9
- Kante, 9
- Kegel, 4
 - dualer, 6
 - Lorentz-, 10
 - ordentlicher, 4
 - polyedrischer, 5
 - punktierter, 4
 - Symmetriegruppe eines, 6
 - Voronoi-, 26
- konisches Optimierungsproblem, 10
 - duales, 11
 - kopositives Problem, 12
 - vollständig positives Problem, 12
 - zulässiges, 10
 - strikt, 11
- Matrix, 1
 - \mathcal{COP} -perfekte, 25
 - benachbarte, 45
 - Hauptminor einer, 53
 - kopositive, 1
 - strikt, 7
 - vollständig positive, 1
 - \mathcal{CP} -Faktorisierung einer, 29
- minimaler Vektor, 25
- Minimum, 25
 - arithmetisches, 25
 - kopositives, 25
- Nichtnegativer Orthant, 3
- perfekte quadratische Form, 26
- Pivotregel, 44
- QCQP, 15
- Ryshkov-Polyeder, 26
- Seite, 9
 - von R induzierte, 23
 - offenliegende, 9
- Simplex, 9
 - Partitionierung eines, 66
 - Standard-, 12
- SOCP, 10
- Standard-Quadratische-Problem, 12
- Strahl, 5
 - Extrem-, 5

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt und ohne fremde Hilfe verfasst habe. Dazu habe ich keine außer den von mir angegebenen Hilfsmitteln und Quellen verwendet und die den benutzten Werken inhaltlich und wörtlich entnommenen Stellen habe ich als solche kenntlich gemacht.

Rostock, den 11.11.2020

Valentin Florian Dannenberg