



**FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA**

Implementierung eines Verfahrens für Matrixspiele mit vektorieller Auszahlung

Bachelorarbeit zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)
im Studiengang Wirtschaftsmathematik

Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik

vorgelegt von Sarah Jonas
Matrikelnummer 185354
geb. am 30.10.2000 in Damme

Erstgutachter: Prof. Dr. Andreas Löhne
Zweitgutachter: Dr. rer. nat. Benjamin Weißing
Jena, den 22.03.2023.

Kurzfassung

Diese Arbeit bezieht sich auf den 2018 veröffentlichten Artikel „*A set optimization approach to zero-sum matrix games with multi-dimensional payoffs*“ und dessen Preprint von Andreas Hamel und Andreas Löhne. In dem Artikel wird ein Ansatz zur Lösung von Matrixspielen mit zwei Spieler:innen und mehrdimensionalen Auszahlungen vorgestellt [vgl. 8, S. 369]. Seine Implementierung wird ausschließlich in dem Preprint hergeleitet und numerisch ausgewertet [vgl. 7, Kapitel 4 und 5].

Basierend auf dem Artikel und dem Lösungsansatz werden in dieser Arbeit ein Algorithmus und seine Implementierung präsentiert, die die optimalen Strategien als Lösung des Spiels für die Spieler:innen bestimmen. Die Implementierung verwendet *bensolve tools* [2, 3, 4, 11]. Abschließend wird anhand ausgewählter Beispiele die Implementierung mit der Originalen [7] verglichen. Die angefertigte Implementierung ist übersichtlich und erzielt korrekte Ergebnisse. Allerdings ist die Laufzeit schlechter als bei der ursprünglichen Implementierung.

Inhaltsverzeichnis

1	Einleitung	4
2	Strategische Spiele	7
2.1	Einführendes	7
2.2	Beispiel und Definition	7
2.3	Gemischte Strategien	9
3	Lösungskonzept basierend auf Mengenrelationen	11
3.1	Minimale und maximale Strategien	12
3.2	Gleichgewichte	14
3.3	Beziehung zum Shapley-Gleichgewicht und POSS	17
4	Implementierung optimaler Strategien	19
4.1	Implementierungsdetails	24
5	Numerische Ergebnisse	25
6	Diskussion	30
7	Zusammenfassung	32
	Literaturverzeichnis	34
	Verzeichnis der Abbildungen	35
	Verzeichnis der Tabellen	36
	Verzeichnis der Algorithmen	37
	Selbstständigkeitserklärung	38
	Anlagen	39

1 Einleitung

„Denn, um es endlich einmal herauszusagen, der Mensch spielt nur, wo er in voller Bedeutung des Wortes Mensch ist, und er ist nur da ganz Mensch, wo er spielt“ (Friedrich Schiller, 1795, [16] 15. Brief, S.88).

Die Spieltheorie beschäftigt sich mit der Vorhersage des Verhaltens der Teilnehmer:innen eines Spiels [vgl. 14, S. xiv]. Inzwischen existiert eine Vielzahl an Lösungsansätzen, die die zukünftigen Aktionen der Spieler:innen ermitteln. Als Beispiel lässt sich das Konzept des Nash-Gleichgewichtes nennen [vgl. 15, S. 14]. Allerdings wird bei den Lösungskonzepten nicht immer Rücksicht auf die Art und Weise genommen, wie die jeweiligen Lösungen erreicht werden. Für eine erfolgreiche Implementierung ist ebendieser Prozess jedoch wesentlich. Algorithmen, die Gleichgewichte oder Lösungen berechnen, bilden den ältesten Teil der algorithmischen Spieltheorie. Sie sind insbesondere für die wirtschaftswissenschaftliche Debatte über die Aussagekraft der mathematisch ermittelten Lösungen bedeutsam. Die effiziente Berechenbarkeit von Lösungen hat sich als überaus wünschenswertes Merkmal für neue und bestehende Lösungskonzepte herausgestellt. Als ein Beispiel lässt sich der Lemke-Howson Algorithmus nennen. Er berechnet für Spiele mit zwei Teilnehmer:innen ein Nash-Gleichgewicht [vgl. 14, S. xiv, 33].

Für die Lösungsansätze ist unter anderem die Dimension der Auszahlungen des Spiels überaus relevant. Alltagsbezogene Spiele mit eindimensionalen Auszahlungen sind beispielsweise Poker oder Auktionen. In einem Pokerspiel kommt es auf den Geldbetrag an, den der oder die Spieler:in schlussendlich erhält und bei einer Auktion liegt die Auszahlung in Form des zu erwerbenden Gegenstandes vor. Für Spiele dieser Art existiert ein bekanntes Lösungskonzept - die Minimax- und Maximin-Strategien. Sie erzeugen Strategievorschläge für die Spieler:innen, die den größtmöglichen Schaden minimieren, der durch die Aktionen der Gegner:innen entstehen kann. Sie agieren für die Spieler:innen sozusagen als Absicherungen gegen den Worst-Case [vgl. 7, S. 2].

Allerdings existieren nicht nur Spiele mit eindimensionalen Auszahlungen, sondern auch Situationen, in denen mehrdimensionale Auszahlungen auftreten. Eine mehrdimensionale Auszahlung verkörpert den Umstand, dass mehrere unvergleichbare Komponenten Teil der Auszahlung sind [vgl. 8, S. 370]. Bereits die gängige Redewendung *Äpfel mit Birnen vergleichen* [9] verkörpert diese Ausgangslage. Angenommen, Äpfel und Birnen sind zwei unvergleichbare Güter. Dann handelt es sich bei einer Auszahlung mit beispielsweise zwei Äpfeln und einer Birne um eine zweidimensionale

Auszahlung. Im Allgemeinen ergeben sich dadurch unvergleichbare Auszahlungen. Hat eine spielende Person die Wahl zwischen zum Beispiel drei Äpfeln und vier Birnen oder vier Äpfeln und drei Birnen, wird sie sich nicht entscheiden können, da der Unterschied zwischen den beiden Auswahlmöglichkeiten nicht quantifizierbar ist [vgl. 8, S. 370].

Es existieren mehrere Ansätze zum Lösen mehrdimensionaler Spiele. Als Beispiele lassen sich die Shapley-Gleichgewichte oder die Pareto optimalen Sicherheitsstrategien (POSS) aufführen. Bei Ersteren fehlen Worst-Case-Abschätzungen und es liegt keine Austauschbarkeit vor. Das bedeutet, dass die beste Antwort der spielenden Person auf ein Shapley-Gleichgewicht nicht unbedingt die beste Antwort auf ein anderes sein muss. Dadurch bleibt unklar, welche Strategie schlussendlich gespielt werden sollte. POSS schützen die Spieler:innen vor ihrem Worst-Case-Szenario bezüglich der einzelnen Bestandteile der Auszahlungen. Dabei werden sehr konservative Strategien erzeugt, die in einigen Fällen wenig sinnvoll sind. Die beschriebenen Schwierigkeiten der beiden Konzepte lassen sich darauf zurückführen, dass das Infimum¹ und das Supremum² in Bezug auf Vektorordnungen in den meisten Fällen nutzlos sind. Entweder existieren sie nicht oder erzeugen „Idealpunkte“, die im Allgemeinen nicht realisierbare Auszahlungen darstellen [vgl. 8, S. 370ff].

Diese Arbeit fokussiert sich auf einen Lösungsansatz von Hamel und Löhne, der das beschriebene mathematische Problem unter Verwendung von Mengenrelationen löst. Durch die vorgeschlagene Mengenrelation und die damit verbundene Mengenoptimierung werden das Infimum und das Supremum wieder eingeführt, weshalb eine vollständige Analogie zu den Minimax- und Maximin-Strategien erzielt werden kann [vgl. 8, S. 372]. Der Ansatz wurde 2018 in dem Artikel *„A set optimization approach to zero-sum matrix games with multi-dimensional payoffs“* eingeführt und löst strategische Spiele mit mehrdimensionalen Auszahlungen. Er liefert mehrere uneindeutige Handlungsempfehlungen für die Spieler:innen, welche Strategie optimalerweise gespielt werden sollte. Im eindimensionalen Fall entspricht er dem Konzept der Minimax- und Maximin-Strategien von John von Neumann [vgl. 8, S. 396].

Ein Algorithmus, der auf dem Lösungsansatz von Hamel und Löhne basiert und eine gegebene Strategie auf Optimalität testet, wird in den unveröffentlichten Teilen des Artikels hergeleitet und seine Implementierung numerisch ausgewertet [vgl. 7, Kapitel 4 und 5]. Der Algorithmus und seine Implementierung stellen im Rahmen der Spieltheorie und der algorithmischen Spieltheorie eine wertvolle Ergänzung für

¹Das Infimum bezeichnet die untere Grenze einer beschränkten Menge [vgl. 5, S. 578].

²Das Supremum bezeichnet die obere Grenze einer beschränkten Menge [vgl. 5, S. 1079].

den Lösungsansatz dar. Des Weiteren ist es denkbar, dass sie für weiterführende Forschungen an Spielen mit mehrdimensionalen Auszahlungen hilfreich sein werden. Diese Bachelorarbeit soll eine übersichtliche, elegante und allgemein nutzbare Implementierung des bereits existierenden Algorithmus aus [7] unter Verwendung der *bensolve tools* [2, 3, 4, 11] hervorbringen. Zudem sollen die Ergebnisse aus [7, Kapitel 5] reproduziert werden. Bei *bensolve tools* [2, 3, 4, 11] handelt es sich um eine freie Software, die beispielsweise konvexe Polyeder berechnen oder verschiedene Optimierungsprobleme lösen kann [vgl. 3]. Für die Umsetzung der Implementierung wird *bensolve tools* [2, 3, 4, 11] gewählt, da die Software unter anderem den VLP-Solver *bensolve* [12, 10] benutzt und gleichzeitig zahlreiche weitere Funktionen besitzt, die insbesondere für die Implementierung im Rahmen dieser Arbeit nützlich erscheinen [vgl. 3]. Durch [7] ist bereits bekannt, dass der Algorithmus mit *bensolve* [12, 10] erfolgreich implementiert werden kann [vgl. 7, Kapitel 5], weshalb nahe liegt, dass die erfolgreiche Entwicklung einer funktionierenden Implementierung unter Verwendung der *bensolve tools* [2, 3, 4, 11] möglich ist.

Die Arbeit ist wie folgt strukturiert. Zunächst werden in Abschnitt 2 strategische Spiele mit ein- und mehrdimensionalen Auszahlungen sowie das Konzept der gemischten Strategien vorgestellt. Danach wird in Abschnitt 3 die Mengenrelation eingeführt, anhand derer im Anschluss minimale und maximale Strategien als optimale Lösungen eines Spiels beschrieben und Gleichgewichtskonzepte definiert werden. Zudem wird der Bezug zu ausgewählten, bereits bestehenden Lösungskonzepten hergestellt, um die Vor- und Nachteile von optimalen Strategien aufzuzeigen. Der Abschnitt 4 befasst sich mit zusätzlich benötigten, mathematischen Aussagen für eine Implementierung des Lösungsansatzes nach Hamel und Löhne sowie mit weiteren Implementierungsdetails. Anschließend werden in Abschnitt 5 mehrere Beispiele behandelt, die die numerischen Ergebnisse der ausgearbeiteten Implementierung zusammenfassen. Insbesondere soll durch die Rekonstruktion von Resultaten aus der bisherigen Forschung in [7, Kapitel 5] eine Beurteilung ermöglicht werden, ob die Verwendung von *bensolve tools* [2, 3, 4, 11] für die Implementierung möglich und vorteilhaft ist. Eine Zusammenfassung der Arbeit und ein Ausblick auf mögliche weiterführende Forschungen bezüglich der Thematik befinden sich in Abschnitt 7.

2 Strategische Spiele

2.1 Einführendes

Spieltheorie ist die Bezeichnung dafür, dass mathematische Hilfsmittel verwendet werden, um Situationen interaktiver Entscheidungsfindung zu modellieren und zu analysieren. In den Situationen sind mehrere Entscheidungsträger:innen mit unterschiedlichen Zielen vertreten und die Entscheidungen jeder einzelnen spielenden Person beeinflusst das Ergebnis aller Spieler:innen. Ziel der Spieltheorie ist es, das Verhalten der Spieler:innen zu antizipieren. Dabei werden teilweise Handlungsempfehlungen erzeugt, die angeben, wie einzelne Spieler:innen ihre jeweiligen Ziele verwirklichen können [vgl. 13, S. xxiii]. Die Vorschläge werden als Lösungen des Spiels bezeichnet [vgl. 20, S. 2]. Die Spieltheorie wurde von John von Neumann und Oskar Morgenstern 1944 im Buch „*Theory of Games and Economic Behavior*“ vorgestellt und findet heute Anwendung in einer Vielzahl von verschiedensten Bereichen, wie beispielsweise in der Volkswirtschaftstheorie, den Politikwissenschaften [vgl. 13, S. xxiii], der Krebsforschung [vgl. 1, S. 1833] oder bei bestimmten Problemen drahtloser Netzwerke [vgl. 6, S. 1, 14].

2.2 Beispiel und Definition

	Kopf	Zahl
Kopf	−1	1
Zahl	1	−1

Tabelle 1: Matching Pennies, abgewandelt aus [19, S. 94, 111]

Die Situation, welche in Tabelle 1 zusammengefasst ist, sei die Folgende: Zwei Personen wählen gleichzeitig und unabhängig voneinander entweder „Kopf“ oder „Zahl“. Weichen die Entscheidungen voneinander ab, zahlt Person *I* eine Geldeinheit an Person *II*; sind sie identisch, erhält *I* die Geldeinheit von *II* [z.B. vgl. 19, S. 111]. Jede Person interessiert sich ausschließlich dafür, wie viele Geldeinheiten er oder sie schlussendlich erhält [vgl. 15, S. 17]. Die beiden Personen werden als Spieler:innen bezeichnet. Die Auswahlmöglichkeiten „Kopf“ und „Zahl“ beschreiben die Menge an möglichen Strategien der Spielenden. Die Geldeinheit, die von der einen Person an die andere übergeben wird, heißt Auszahlung und

$$A = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.1)$$

ist die Auszahlungsmatrix [z.B. vgl. 13, 20, S. 52, 113, S. 107f]. A wird als Verlustmatrix für Person I interpretiert. Das heißt, Person I übergibt den entsprechenden Matrixeintrag a_{ij} an die andere Person. Liegt ein negativer Matrixeintrag vor, bedeutet dies, dass Person I den besagten Betrag von Person II erhält [vgl. 8, S. 375]. In der eben beschriebenen Beispielsituation aus Tabelle 1 zieht Person I eine kleinere Auszahlung gegenüber einer größeren vor, weil dadurch ein geringerer Verlust für sie einhergeht. Entgegengesetzt dazu, favorisiert Person II die größere Auszahlung. Bei der aufgeführten Situation der *Matching Pennies* handelt es sich um ein sogenanntes strategisches Spiel [vgl. 15, S. 15ff].

Ein Spiel ist ein mathematisches Modell einer Situation interaktiver Entscheidungsfindung, in der jede spielende Person danach strebt, ihr bestmögliches Ergebnis zu erzielen, in dem Wissen, dass alle anderen Spieler:innen das Gleiche erzielen wollen [vgl. 13, S. 9]. Ein Spiel heißt endlich, wenn die Menge der möglichen Strategien jeglicher Spieler:innen endlich ist. Kann ein Spiel durch eine Matrix beschrieben werden, wie beispielsweise *Matching Pennies* durch die Matrix A in (2.1), wird es Matrixspiel genannt [z. B. vgl. 13, S. 77f].

Im Rahmen dieser Arbeit werden ausschließlich endliche Matrixspiele mit zwei Teilnehmer:innen betrachtet: Spieler I und Spielerin II [vgl. 8, S. 375]. Zudem werden nicht nur Spiele mit eindimensionalen Auszahlungen behandelt, wie im Beispiel *Matching Pennies*, sondern insbesondere Spiele mit Auszahlungen höherer Dimensionen betrachtet. Die mehrdimensionalen oder auch vektoriellen Auszahlungen sind jeweils Vektoren des \mathbb{R}^K [vgl. 13, S. 569f]. Spiele mit mehrdimensionalen Auszahlungen verkörpern Situationen, in denen die Ergebnisse der Interaktionen der Spieler:innen aus mehreren unvergleichbaren Größen bestehen [vgl. 13, S. 569]. Hierbei kann es sich zum Beispiel um verschiedene Vermögenswerte [vgl. 8, S. 375] oder Arbeitsverträge mit mehreren unvergleichbaren Komponenten (Höhe des Gehalts, Anzahl der Urlaubstage) handeln [vgl. 13, S. 569]. Bezüglich der Präferenzen der Spieler:innen wird im Folgenden ausschließlich angenommen, dass Spieler I im Allgemeinen weniger Verlust und Spielerin II mehr Gewinn präferiert [vgl. 8, S. 376]. Für den mehrdimensionalen Fall bedeutet dies, dass Spieler I die einzelnen Elemente des Auszahlungsvektors verkleinern will, während Spielerin II das genaue Gegenteil realisieren möchte, indem sie eine Erhöhung der jeweiligen Komponenten anstrebt [vgl. 17, S. 59]. Ihre Präferenzen können durch die Relation $\leq_{\mathbb{R}_+^K}$ repräsentiert werden. Die Halbordnung $\leq_{\mathbb{R}_+^K}$ ist definiert in (2.2) [vgl. 8, S. 373, 376].

$$x, y \in \mathbb{R}^K : y \leq_{\mathbb{R}_+^K} x \Leftrightarrow x - y \in \mathbb{R}_+^K \quad (2.2)$$

Des Weiteren werden im Folgenden ausschließlich Auszahlungsmatrizen der Form $A = (a_{ij})_{m \times n}$ mit

$$a_{ij} = (a_{ij}^1, a_{ij}^2, a_{ij}^3, \dots, a_{ij}^K)^T \in \mathbb{R}^K$$

verwendet, um die Spiele darzustellen. Im Rahmen dieser Arbeit werden reine und gemischte Strategien von den Spieler:innen verwendet. Die gespielten Strategien werden als rein bezeichnet, wenn Spieler I und Spielerin II jeweils unabhängig voneinander genau eine einzige Zeile $i \in \{1, \dots, m\}$ beziehungsweise Spalte $j \in \{1, \dots, n\}$ der Matrix A auswählen [z. B. vgl. 8, S. 375].

2.3 Gemischte Strategien

Bei dem Spiel *Matching Pennies* ist jede Art zu Spielen genauso gut wie irgendeine andere. Per se liegt in den einzelnen Strategien „Kopf“ oder „Zahl“ kein Vor- oder Nachteil [vgl. 19, S. 111]. Die Lösung liegt im Mischen der Strategien [vgl. 19, S. 144].

„Die Strategie des Spielers besteht weder in der Wahl von „Kopf“ noch von „Wappen“, sondern in der Wahl von „Kopf“ mit der Wahrscheinlichkeit $1/2$ und der Wahl von „Wappen“ mit der Wahrscheinlichkeit $1/2$ “ (Von Neumann und Morgenstern, 1947, [18] S.146).

Bei Spielen mit sogenannten **gemischten Strategien** werden die Zeile i und die Spalte j der $m \times n$ - Auszahlungsmatrix A von den jeweiligen Spieler:innen mit der Wahrscheinlichkeit p_i und q_j ausgewählt. Daraus ergeben sich die Mengen

$$P = \{p \in \mathbb{R}_+^m \mid \sum_{i=1}^m p_i = 1\} \text{ und } Q = \{q \in \mathbb{R}_+^n \mid \sum_{j=1}^n q_j = 1\},$$

die sämtliche zulässige Strategien p und q für Spieler I , beziehungsweise Spielerin II , enthalten [z. B. vgl. 13, S. 146f]. Eine gemischte Strategie einer spielenden Person bezeichnet also eine Wahrscheinlichkeitsverteilung über die Menge der reinen Strategien [z. B. vgl. 20, S. 139ff]. Falls die Strategien $p \in P$ und $q \in Q$ gespielt werden, beschreibt der Vektor

$$v(p, q) = \sum_{i=1}^m \sum_{j=1}^n p_i a_{ij} q_j \in \mathbb{R}^K$$

den erwarteten Verlust für Spieler I [vgl. 13, S. 576]. Es wird definiert, dass Spieler I Strategie $p^* \in P$ gegenüber $p \in P$ **präferiert**, falls für alle gegnerischen Strategien $q \in Q$ gilt, dass

$$v(p^*, q) \leq_{\mathbb{R}_+^K} v(p, q). \quad (2.3)$$

Spieler I bevorzugt also diejenige Strategie, welche einen kleineren erwarteten Verlust generiert. Analog präferiert Spielerin II jene Strategie $q \in Q$, die den größeren erwarteten Gewinn hervorbringt [vgl. 8, S. 376].

Beispiel 2.1. Bei

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

handelt es sich um eine 2×2 Spielmatrix mit zweidimensionaler Auszahlung. Spieler I wählt die erste Zeile der Matrix A mit Wahrscheinlichkeit p_1 und die zweite Zeile mit Wahrscheinlichkeit p_2 aus, wobei die Summe der Wahrscheinlichkeiten gleich 1 ist. Insgesamt ergibt sich für Spieler I die Strategie $p = (p_1, p_2)^T$. In diesem Fall ist die Menge P äquivalent zu dem Intervall des \mathbb{R}^2 , das die Punkte $(1, 0)$ und $(0, 1)$ verbindet [vgl. 13, S. 146]. Gleiches gilt für die Spielerin II , die sich zwischen den Spalten der Matrix entscheidet. Seien die Strategien

$$\bar{p} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \text{ und } \bar{q} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

gegeben. Dann erwarten Spieler I einen Verlust und Spielerin II einen Gewinn in Höhe von

$$v(\bar{p}, \bar{q}) = \sum_{i=1}^2 \sum_{j=1}^2 \bar{p}_i a_{ij} \bar{q}_j = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

Wählt Spieler I die Strategie $p \in P$, liegt seine erwartete Auszahlung, unabhängig von der Entscheidung seiner Gegnerin, in

$$v_I(p) = \{v(p, q) | q \in Q\}.$$

Nach [13, S. 576] und [8, S.377] handelt es sich bei $v_I(p)$ um einen Polyeder des \mathbb{R}^K der Form

$$v_I(p) = co \left\{ \sum_{i=1}^m p_i a_{i1}, \dots, \sum_{i=1}^m p_i a_{in} \right\} \subseteq \mathbb{R}^K.$$

$$v_{II}(q) = \{v(p, q) | p \in P\} = co \left\{ \sum_{j=1}^n q_j a_{1j}, \dots, \sum_{j=1}^n q_j a_{mj} \right\} \subseteq \mathbb{R}^K$$

ergibt sich analog für die Spielerin II mit der Strategie $q \in Q$ [vgl. 13, S. 576].

3 Lösungskonzept basierend auf Mengenrelationen

Das Konzept von Hamel und Löhne ermöglicht schlüssige und berechenbare Worst-Case-Abschätzungen für Matrixspiele mit mehrdimensionalen Auszahlungen. Es erzeugt Strategiepaare, die gespielt werden sollten, wenn die Spieler:innen „verlustavers³“ sind und nichts über ihre Präferenzen bekannt ist außer der Tatsache, dass sie „weniger Verlust“ und „mehr Gewinn“ bevorzugen [vgl. 8, S. 396]. Die nachfolgend definierte Mengenrelation aus [8, S. 373, 377] ist die Basis des Lösungsansatzes. Für $p, \bar{p} \in P$ und $q, \bar{q} \in Q$ ist

$$\begin{aligned} v_I(\bar{p}) \preceq v_I(p) &: \Leftrightarrow v_I(p) \subseteq v_I(\bar{p}) - \mathbb{R}_+^K \\ v_{II}(\bar{q}) \preceq v_{II}(q) &: \Leftrightarrow v_{II}(q) \subseteq v_{II}(\bar{q}) + \mathbb{R}_+^K. \end{aligned} \quad (3.1)$$

Proposition 3.1. *Sei $p, p^* \in P$. Spieler I präferiert p^* gegenüber p , falls $v_I(p^*) \preceq v_I(p)$.*

Beweis. Die Aussage folgt aus den Annahmen über die Präferenzen der Spieler:innen, der Definitionen der Präferenzrelation „ $\leq_{\mathbb{R}_+^K}$ “ sowie der Definition der Mengenrelation „ \preceq “ [vgl. 8, S. 377]. \square

Analog präferiert Spielerin II q^* gegenüber q , falls $v_{II}(q^*) \preceq v_{II}(q)$. Die Menge

$$V_I(p) := v_I(p) - \mathbb{R}_+^K$$

repräsentiert alle möglichen Verluste für Spieler I , die kleiner oder gleich dem erzeugten Verlust von Strategie p sind. Aus der Definition ergibt sich die Tatsache, dass Spieler I die Strategie p^* gegenüber p präferiert, falls $V_I(p^*) \subset V_I(p)$. Das bedeutet, dass diejenige Strategie favorisiert wird, welche eine kleinere Teilmenge an potenziellen, besseren Ergebnissen aufweist.

$$V_{II}(q) := v_{II}(q) + \mathbb{R}_+^K$$

beinhaltet für Spielerin II alle möglichen Gewinne, die größer oder gleich $v_{II}(q)$ sind. Auch hier gilt, dass die Strategie q^* gegenüber der Strategie q von Spielerin II bevorzugt wird, falls $V_{II}(q^*) \subset V_{II}(q)$ [vgl. 8, S. 378f].

³Aversion bedeutet Abneigung oder Widerwille [vgl. 5, S. 237].

Algorithmus 1 Pseudocode Berechnung von $V_I(p)$

- 1: Berechne die Werte in der konvexen Hülle von $v_I(p)$
 - 2: Erzeuge deren konvexe Hülle
 - 3: Subtrahiere von der Menge den Kegel \mathbb{R}_+^K
-

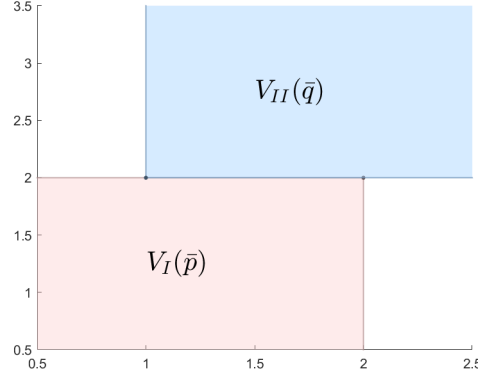


Abbildung 1: Mengen $V_I(\bar{p})$ und $V_{II}(\bar{q})$ für die Strategien und das Spiel aus Beispiel 2.1.

3.1 Minimale und maximale Strategien

Im Allgemeinen ist das Ziel des Spielers I eine zulässige Strategie $p \in P$ zu spielen, sodass sein resultierender Verlust möglichst gering ist. Das kann erreicht werden, indem eine Strategie gewählt wird, die den maximal erwarteten Verlust minimiert. Dadurch werden parallel alle möglichen Verluste minimiert. Da der größtmögliche Verlust ein Teil der Menge $V_I(p)$ ist, kann er durch die Verkleinerung von $V_I(p)$ verringert werden. Dementsprechend ist eine maximale Verkleinerung von $V_I(p)$ bezüglich der Halbordnung „ \subseteq “ äquivalent zu einer maximalen Minimierung des größtmöglichen Verlustes. Eine Strategie, die ebendiese Minimierung garantiert, ist eine minimale Strategie. Eine Strategie $\tilde{p} \in P$ heißt **minimal** für den Spieler I , wenn

$$\nexists p \in P : V_I(p) \subseteq V_I(\tilde{p}) \text{ und } V_I(p) \neq V_I(\tilde{p}).$$

Minimale Strategien fungieren somit als eine Absicherung gegen den schlimmsten Fall: den größtmöglichen Verlust. $\text{MIN}(I)$ bezeichnet die Menge aller minimalen Strategien für Spieler I [vgl. 8, S. 378f]. Analog ist für Spielerin II die Menge aller **maximalen Strategien** definiert durch

$$\text{MAX}(II) = \{\tilde{q} \in Q \mid \nexists q \in Q : V_{II}(q) \subseteq V_{II}(\tilde{q}) \text{ und } V_{II}(q) \neq V_{II}(\tilde{q})\}.$$

Eine Strategie heißt **optimal** für die Spieler:innen I und II , wenn sie minimal beziehungsweise maximal ist. Da minimale und maximale Strategien unabhängig von den Aktionen des Gegners beziehungsweise der Gegnerin sind, gelten sie als austauschbar. Das bedeutet, dass die beiden minimalen oder maximalen Strategien von zwei gegebenen optimalen Strategiepaaren jeweils miteinander vertauscht werden können, ohne dass die neuen Strategiepaare ihre Optimalität verlieren [vgl. 8, S. 378f].

Proposition 3.2. *Die Spieler:innen präferieren eine optimale Strategie gegenüber einer nicht optimalen Strategie.*

Beweis. Die Aussage folgt aus der Proposition 3.1 und der Definition optimaler Strategien. \square

Die Proposition 3.3 sagt aus, dass für die Spieler:innen immer optimale Strategien existieren [vgl. 8, S 381].

Proposition 3.3. [8, S. 381 Theorem 3.8] *Für alle Strategiepaare $(p, q) \in P \times Q$ existiert ein optimales Strategiepaar $(\tilde{p}, \tilde{q}) \in \text{MIN}(I) \times \text{MAX}(II)$ mit*

$$V_I(\tilde{p}) \subseteq V_I(p) \text{ und } V_I(\tilde{p}) \neq V_I(p).$$

Beweis. Siehe [8, S. 381]. \square

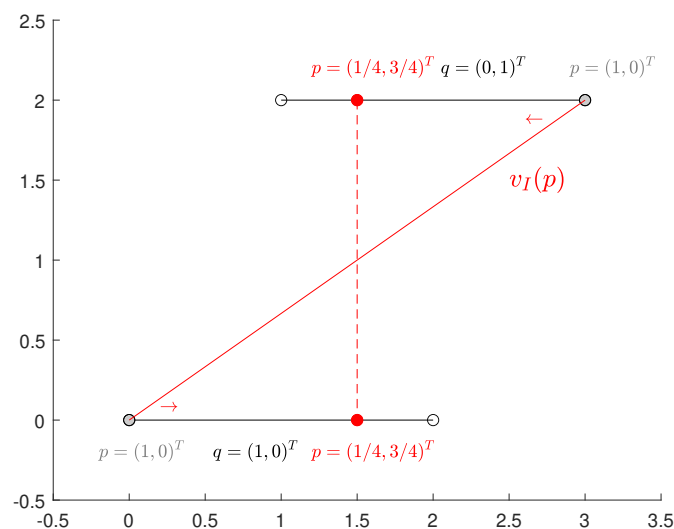


Abbildung 2: Grafische Ermittlung der Menge der minimalen Strategien für das Spiel aus Beispiel 2.1.

Beispiel 3.4. Für Matrix A aus Beispiel 2.1 ergeben sich folgende Mengen der minimalen bzw. maximalen Strategien:

$$\text{MIN}(I) = \left\{ p \in P \mid 0 \leq p_1 \leq \frac{1}{4} \right\} \quad \text{und} \quad \text{MAX}(II) = \left\{ q \in Q \mid 0 \leq q_1 \leq \frac{1}{2} \right\}.$$

In Abbildung 2 ist die grafische Ermittlung von $\text{MIN}(I)$ aufgezeigt, damit die Lösung unmittelbar nachvollzogen werden kann. Die schwarzen Linien beschreiben alle realisierbaren Auszahlungen, falls Spielerin II entweder die erste oder die zweite Spalte wählt. An den jeweiligen Enden der Linien (schwarze und graue Kreise) befinden sich die Beträge, die Spieler I bezahlen muss, wenn er Zeile eins oder zwei aussucht, unter der Annahme, dass seine Gegnerin sich vollständig für eine der beiden Spalten entscheidet. Die eingezeichnete rote Linie repräsentiert die Menge aller möglichen Auszahlungen $v_I(p)$, insofern der Spieler I die erste Zeile strikt favorisiert. Werden die Endpunkte der roten Linie entlang der schwarzen Linie in Richtung der Pfeile verschoben, erhöht sich die Wahrscheinlichkeit für die Wahl der zweiten Zeile der Auszahlungsmatrix A . Die Zunahme von p_2 erfolgt in gleichem Maße wie die Abnahme von p_1 , da $p_1 + p_2 = 1$. Bei der stetigen Verschiebung von $v_I(p)$ fällt auf, dass für alle

$$p_1 > \bar{p}_1 \geq \frac{1}{4} : V_I(\bar{p}) \subset V_I(p). \quad (3.2)$$

Demzufolge wird die Menge an möglichen geringeren Verlusten eindeutig durch die Vergrößerung von p_2 verkleinert. Die rot gestrichelte Linie kennzeichnet dabei die Grenze. Ab hier führt jede weitere Verkleinerung von p_1 zu einer anderen minimalen Strategie, ohne $V_I(p)$ strikt zu reduzieren. Analog kann die Berechnung von $\text{MAX}(II)$ veranschaulicht werden. Zusätzlich wird (3.2) durch die Animation in Abschnitt 7 verdeutlicht. In der Animation ist die Menge $V_I(p)$ in Abhängigkeit von der Strategie p aufgezeigt. Dabei ist $p_1 \in \{1, \frac{7}{8}, \frac{3}{4}, \frac{5}{8}, \frac{1}{2}, \frac{3}{8}, \frac{1}{4}, \frac{1}{8}, 0\}$. p_1 wird somit beginnend bei 1 schrittweise um den Wert $\frac{1}{8}$ verringert. Die einzelnen Bilder der Animation sind auf <https://github.com/SarahStudent/Bachelorarbeit> abrufbar.

3.2 Gleichgewichte

Für die nachfolgenden Definitionen werden das Maximum der Menge $v_I(p)$ und das Minimum von $v_{II}(q)$ benötigt. $\text{Max } v_I(p)$ beschreibt die maximal erwartete Auszahlung, wenn von Spieler I die Strategie p gespielt wird, und stellt somit das Worst-Case-Szenario für Spieler I dar. Analog lässt sich $\text{Min } v_{II}(q)$ als Worst-Case für Spielerin I interpretieren, wenn sie die Strategie q spielt [vgl. 8, S. 377f].

Beispiel 3.5. Für die Auszahlungsmatrix aus Beispiel 2.1 ist die Strategie $\bar{p} = (1, 0)^T$ nicht minimal (siehe Beispiel 3.4). Die maximale Auszahlung, die unter \bar{p} erzielt werden kann, ist

$$\text{Max } v_I(\bar{p}) = \begin{pmatrix} 3 \\ 2 \end{pmatrix}.$$

Für die minimale Strategie $\hat{p} = (0.25, 0.75)^T$ gilt $V_I(\hat{p}) \subset V_I(\bar{p})$ und die Strategie garantiert eine maximal mögliche Auszahlung in Höhe von

$$\text{Max } v_I(\hat{p}) = \begin{pmatrix} 2 \\ \frac{3}{2} \end{pmatrix}.$$

Spieler I kann seinen größtmöglichen, erwarteten Verlust verkleinern, indem er von der Strategie \bar{p} zur minimalen Strategie \hat{p} wechselt. Ein weiterer Wechsel zu einer anderen minimalen Strategie wird die maximal erwartete Auszahlung nicht strikt verbessern, sondern lediglich eine neue Abschätzung liefern, die mit $\text{Max } v_I(\hat{p})$ unvergleichbar ist.

Ein Strategiepaar $(\tilde{p}, \tilde{q}) \in P \times Q$ heißt **Shapley-Gleichgewicht**, falls

$$v(\tilde{p}, \tilde{q}) \in \text{Max } v_I(\tilde{p}) \cap \text{Min } v_{II}(\tilde{q})$$

und **starkes Shapley-Gleichgewicht**, wenn

$$V_I(\tilde{p}) \cap V_{II}(\tilde{q}) = \text{Max } v_I(\tilde{p}) \cap \text{Min } v_{II}(\tilde{q})$$

erfüllt ist [vgl. 8, S. 382 Def. 3.11]. Ein starkes Shapley-Gleichgewicht genügt immer der Definition eines Shapley-Gleichgewichtes. Die umgekehrte Implikation hingegen ist inkorrekt: Nicht jedes Shapley-Gleichgewicht ist ein starkes [vgl. 8, S. 382].

Beispiel 3.6. Betrachtet wird das Spiel aus Beispiel 2.1. Für das Strategiepaar

$$(\bar{p}, \bar{q}) = \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \right)$$

lässt sich nachprüfen, dass ein starkes Shapley-Gleichgewicht vorliegt. Das Paar

$$(\bar{\bar{p}}, \bar{\bar{q}}) = \left(\begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

ist kein Shapley-Gleichgewicht.

Bei einem Shapley-Gleichgewicht werden Auszahlungen erzielt, die bezüglich der gewählten Strategien von keinem der Spieler:innen verbessert werden können. Ein starkes Shapley-Gleichgewicht hingegen generiert Auszahlungen, die in Bezug auf die Abschätzung des ungünstigsten Falls – dem größtmöglichen Verlust oder dem kleinstmöglichen Gewinn – nicht vorteilhafter gemacht werden können. Ein Strategiepaar $(\tilde{p}, \tilde{q}) \in P \times Q$ wird als ein **Mengenrelationsgleichgewicht** bezeichnet, falls es sich bei \tilde{p} um eine minimale und bei \tilde{q} um eine maximale Strategie handelt. Ein Mengenrelationsgleichgewicht heißt **Shapley-Mengen-Gleichgewicht**, wenn parallel ein Shapley-Gleichgewicht vorliegt. Ist es darüber hinaus ein starkes Shapley-Gleichgewicht, ergibt sich ein **starkes Shapley-Mengen-Gleichgewicht**. Analog zu dem Konzept der Shapley-Gleichgewichte gilt auch hier, dass nicht jedes Shapley-Mengen-Gleichgewicht ein starkes ist. Des Weiteren hat jedes Spiel mit vektorieller Auszahlung ein starkes Shapley-Mengen-Gleichgewicht [vgl. 8, S. 382ff].

Beispiel 3.7. Die Strategien $\bar{p}, \bar{q}, \bar{\bar{p}}$ und $\bar{\bar{q}}$ aus dem Beispiel 3.6 sind jeweils minimal beziehungsweise maximal für das Spiel in 2.1. Dementsprechend sind beide Strategiepaare Mengenrelationsgleichgewichte. Da (\bar{p}, \bar{q}) zusätzlich ein starkes Shapley-Gleichgewicht ist, handelt es sich um ein starkes Shapley-Mengengleichgewicht. In $(\bar{\bar{p}}, \bar{\bar{q}})$ liegt kein Shapley-Mengen-Gleichgewicht vor.

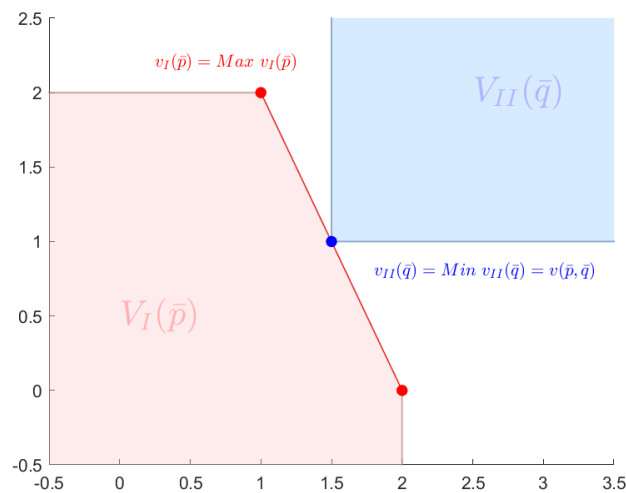


Abbildung 3: Visualisierung der zu prüfenden Größen bei einem Shapley-Gleichgewicht für (\bar{p}, \bar{q}) aus Beispiel 3.6.

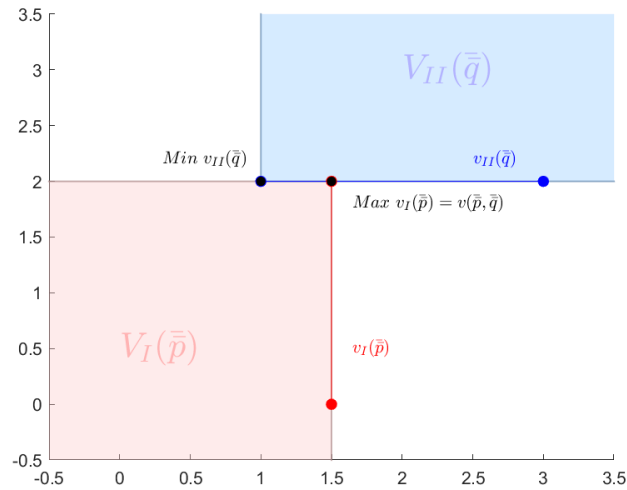


Abbildung 4: Visualisierung der zu prüfenden Größen bei einem Shapley-Gleichgewicht für (\bar{p}, \bar{q}) aus Beispiel 3.6.

Für die minimalen und maximalen Strategien aus Beispiel 3.6 wird durch die Abbildungen 3 und 4 deutlich, dass die resultierenden Auszahlungen der jeweiligen optimalen Strategiepaare unterschiedlich sind.

$$v(\bar{p}, \bar{q}) = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} \neq v(\bar{\bar{p}}, \bar{\bar{q}}) = \begin{bmatrix} 1.5 \\ 2 \end{bmatrix}$$

Demnach erzeugt das Spielen von optimalen Strategien nicht immer ein und dieselbe Auszahlung - einen sogenannten Gleichgewichtswert. Hier liegt ein wesentlicher Unterschied zu Spielen mit eindimensionalen Auszahlungen vor: im mehrdimensionalen Fall ist Mini- oder Maximalität einer Strategie nicht mit dem Erzielen eines Gleichgewichtswertes äquivalent [vgl. 8, S. 383].

3.3 Beziehung zum Shapley-Gleichgewicht und POSS

Der Begriff des Shapley-Gleichgewichts ist bereits in Unterabschnitt 3.2 definiert. Die einzelnen Strategien p und q eines Shapley-Gleichgewichtes (p, q) heißen **Shapley-Lösungen**. Es lässt sich zeigen, dass eine Strategie nicht zwingend eine Shapley-Lösung sein muss, wenn sie minimal ist und die Strategie nicht automatisch minimal ist, wenn es sich um eine Shapley-Lösung handelt. Demnach sind die beiden Konzepte unabhängig voneinander [vgl. 8, S. 387ff]. Des Weiteren sind Shapley-Gleichgewichtsstrategien im Gegensatz zu minimalen oder maximalen Strategien

nicht austauschbar, da bei ersterem die Strategien beider Spieler:innen verwendet werden [vgl. 8, S. 383].

Betrachtet werden die nachfolgenden Mengen.

$$\begin{aligned} W_I(p) &= \cap_{q \in Q} (v(p, q) + \mathbb{R}_+^K) & \omega_I &= \cup_{p \in P} W_I(p) \\ W_{II}(q) &= \cap_{p \in P} (v(p, q) - \mathbb{R}_+^K) & \omega_{II} &= \cup_{q \in Q} W_{II}(q) \end{aligned}$$

Die Menge $W_I(p)$ beinhaltet alle potenziellen, erwarteten Verluste, falls Spieler I die Strategie p spielt, und ist unabhängig von den Aktionen der Spielerin II . Eine maximale Vergrößerung von $W_I(p)$ erscheint für den Spieler I sinnvoll, weil dadurch die Wahrscheinlichkeit für ihn steigt „kleine“ Verluste in der Menge vorzufinden. Möchte Spieler I seinen maximal erwarteten Verlust minimieren, muss er die Menge ω_I ermitteln. $W_{II}(q)$ umfasst die erwarteten Gewinne der Spielerin II bezüglich der Strategie q . Die erwarteten Gewinne für die Strategie q werden ebenfalls nicht von den gegnerischen Strategien beeinflusst. Analog dazu sollte die Menge $W_{II}(q)$ größtmöglich sein, damit die Wahrscheinlichkeit steigt, dass sich „große“ Gewinne in der Menge befinden. Die Strategien \tilde{p} und \tilde{q} aus den Mengen

$$\text{POSS}(I) = \{\tilde{p} \in P \mid \nexists p \in P : W_I(p) \supseteq W_I(\tilde{p}) \text{ und } W_I(p) \neq W_I(\tilde{p})\}$$

und

$$\text{POSS}(II) = \{\tilde{q} \in Q \mid \nexists q \in Q : W_{II}(q) \supseteq W_{II}(\tilde{q}) \text{ und } W_{II}(q) \neq W_{II}(\tilde{q})\}$$

heißen **pareto-optimale Sicherheitsstrategien (POSS)** für Spieler I beziehungsweise Spielerin II . Aus den Definitionen resultiert, dass für alle Strategien $p \in P$ und $q \in Q$ der Spieler:innen

$$W_{II}(q) \subseteq V_I(p) \quad \text{und} \quad W_I(q) \subseteq V_{II}(p).$$

Dies ist in Abbildung 7 zu erkennen. Ferner lässt sich nachweisen, dass minimale und maximale Strategien keine schlechteren Auszahlungen erzeugen als pareto-optimale Sicherheitsstrategien, sondern teilweise sogar bessere Worst-Case-Abschätzungen für die möglichen Auszahlungen liefern [vgl. 8, S. 392-395]. Für weitere Beziehungen zwischen bestehenden Lösungskonzepten und dem Mengenoptimierungsansatz nach Hamel und Löhne wird auf [8, Kapitel 4] verwiesen.

4 Implementierung optimaler Strategien

Das Verfahren basiert darauf zu überprüfen, ob die zulässige Strategie $p \in P$ für den Spieler I optimal ist. Da unendlich viele zu testende Strategien vorliegen, wird mit einer endlichen Teilmenge an Strategien $\bar{P} \subseteq P$ gearbeitet, um repräsentative Aussagen über die Menge der optimalen Strategien treffen zu können. Spieler I entscheidet sich für eine der optimalen Strategien anhand der endlich vielen Auszahlungsmengen $V_I(\tilde{p})$, die von den optimalen Strategien $\tilde{p} \in \bar{P} \subseteq P$ erzeugt werden. Dabei wird angenommen, dass die endlich vielen optimalen Strategien \tilde{p} die Menge aller optimaler Strategien adäquat repräsentieren und Spieler I sämtliche Informationen der Mengen $V_I(\tilde{p})$ verwenden kann, um schlussendlich eine einzelne optimale Strategie p^* auszuwählen [vgl. 7, S. 20]. Die nachfolgenden beiden Präpositionen 4.1 aus [7, S. 20 Proposition 4.1] und 4.2 aus [vgl. 7, S. 21] dienen dazu die Ausdrücke

$$V_I(p) \subseteq V_I(\bar{p}) \quad \text{und} \quad V_I(p) \neq V_I(\bar{p})$$

als lineare Ungleichungen darzustellen. Anschließend werden die Ungleichungen in Präposition 4.3 verwendet, um ein lineares Optimierungsproblem aufzustellen, welches überprüft, ob die betrachtete Strategie \bar{p} minimal für den Spieler I ist. \bar{p} wird dafür gegen alle anderen möglichen Strategien $p \in P$ getestet [vgl. 7, S. 20].

Proposition 4.1. [vgl. 7, S. 20 Proposition 4.1] Seien eine Strategie $\bar{p} \in P$, eine Matrix $H \in \mathbb{R}^{s \times m}$ und ein Vektor $h \in \mathbb{R}^s$ gegeben, sodass H und h die H -Darstellung des Polyeders $V_I(\bar{p})$ darstellen.

$$V_I(\bar{p}) = \{y \in \mathbb{R}^K \mid H \cdot y \geq h\}$$

Für eine beliebige Strategie $p \in P$ gilt, dass für alle $j = 1, \dots, n$

$$V_I(p) \subseteq V_I(\bar{p}) \Leftrightarrow H \cdot \left(\sum_{i=1}^m p_i g_{ij} \right) \geq h.$$

Beweis. Für alle $p \in P$ ist

$$V_I(p) = \text{co} \left\{ \sum_{i=1}^m p_i g_{i1}, \dots, \sum_{i=1}^m p_i g_{in} \right\} - \mathbb{R}_+^K.$$

Infolgedessen ist $V_I(p)$ eine Teilmenge von $V_I(\bar{p})$ genau dann, wenn jeder Punkt, der die konvexe Hülle definiert, in $V_I(\bar{p})$ liegt [vgl. 7, S. 20f]. \square

Proposition 4.2. Sei die Strategie $\bar{p} \in P$ und für jede Ecke y^l ($l = 1, \dots, r$) des Polyeders $V_I(\bar{p})$ eine Stützhyperebene der Form

$$H^l = \{y \in \mathbb{R}^K \mid (u^l)^T y = \gamma_l\}$$

gegeben für die gilt, dass

$$V_I(\bar{p}) \cap H^l = \{y^l\} \quad \text{und} \quad (u^l)^T y \geq \gamma_l \text{ für alle } y \in V_I(\bar{p}).$$

Es existieren $l \in \{1, \dots, r\}$, $\varepsilon_l > 0$ und eine Strategie $p \in P$ mit der Eigenschaft $V_I(p) \subseteq V_I(\bar{p})$, sodass für alle $j = 1, \dots, n$

$$(u^l)^T \left(\sum_{i=1}^m p_i g_{ij} \right) \geq \gamma_l + \varepsilon_l, \quad (4.1)$$

genau dann, wenn \bar{p} keine minimale Strategie ist.

Beweis. „ \Rightarrow “: Sei $V_I(p) \subseteq V_I(\bar{p})$, dann ist für alle $j = 1, \dots, n$

$$(u^l)^T \left(\sum_{i=1}^m p_i g_{ij} \right) \geq \gamma_l. \quad (4.2)$$

Gelten zusätzlich (4.1) und ihre Voraussetzungen, ist die Ecke y^l kein Element der Menge $V_I(p)$. Deshalb sind die Mengen $V_I(p)$ und $V_I(\bar{p})$ nicht identisch, woraus folgt, dass \bar{p} keine minimale Strategie ist [vgl. 7, S. 21].

„ \Leftarrow “: Sei (4.1) für alle $l \in \{1, \dots, r\}$, $\varepsilon_l > 0$ und jede Strategie $p \in P$ mit $V_I(p) \subseteq V_I(\bar{p})$ verletzt. Dann folgt aus (4.2), dass für jede dieser Strategien p und jede Ecke y^l von $V_I(\bar{p})$ eine Indexmenge $j_{(P,l)} \in \{1, \dots, n\}$ existiert, sodass

$$(u^l)^T \left(\sum_{i=1}^m p_i g_{ij_{(P,l)}} \right) = \gamma_l. \quad (4.3)$$

Aufgrund der festgelegten Definition und Darstellungsweise der Stützhyperebene H^l ist (4.3) äquivalent zu der Tatsache, dass jede Ecke von $V_I(\bar{p})$ für eine feste Strategie $p \in P$ mit einem Punkt aus $V_I(p)$ zusammenfällt. Demnach sind die Mengen $V_I(p)$ und $V_I(\bar{p})$ identisch für alle Strategien $p \in P$ mit $V_I(p) \subseteq V_I(\bar{p})$, weshalb es sich dann bei der Strategie \bar{p} um eine minimale Strategie handelt [vgl. 7, S. 21]. \square

Proposition 4.3. [vgl. 7, S. 22 Theorem 4.3] Die Strategie $\bar{p} \in P$ ist minimal genau dann, wenn das lineare Optimierungsproblem (4.4) einen Optimalwert gleich null hat.

$$\begin{aligned}
 & \max_{p, \varepsilon} \quad \sum_{l=1}^r \varepsilon_l \tag{4.4} \\
 \text{bezüglich:} \quad & (u^l)^T \left(\sum_{i=1}^m p_i g_{ij} \right) \geq \gamma_l + \varepsilon_l \quad \forall j = 1, \dots, n, \forall l = 1, \dots, r \\
 & H \cdot \left(\sum_{i=1}^m p_i g_{ij} \right) \geq h \quad \forall j = 1, \dots, n \\
 & p \geq 0 \\
 & e^T p = 1
 \end{aligned}$$

Beweis. Die Aussage folgt direkt aus den Propositionen 4.1 und 4.2 [vgl. 7, S. 22]. \square

Algorithmus 2 testet, ob die Strategie $\bar{p} \in P$ minimal ist. Grundlage für das Vorgehen ist das Optimierungsproblem aus Proposition 4.3. Die Berechnung der Stützhyperebenen erfolgt durch eine externe Funktion, welche in Algorithmus 3 beschrieben wird. Im Allgemeinen lassen sich Stützhyperebenen von Ecken eines Polyeders berechnen, indem die Ungleichungsrestriktionen der H-Darstellung, bei denen in der Ecke Gleichheit vorliegt, aufsummiert werden. Aus der Addition der dadurch bestimmten Matrixzeilen ergibt sich der gesuchte Parameter u . γ lässt sich durch das Aufsummieren der entsprechenden Einträge des Vektors h errechnen. Das entsprechende Verfahren für Spielerin II lässt sich analog aufstellen [vgl. 7, S. 22]. Das heißt, dass für die Bestimmung der maximalen Strategien der selbe Test wie für minimale Strategien verwendet werden kann, wenn die Matrix des Spiels transponiert und mit -1 multipliziert wird. Für eine korrekte Darstellung der zugehörigen $V_{II}(q)$ müssen die Mengen erneut mit -1 multipliziert werden. Die ausgearbeitete Implementierung der Algorithmen 1, 2 und 3 befindet sich mit Beispieleingaben in Abschnitt 7.

Algorithmus 2 Pseudocode Optimalitätstest Strategie \bar{p}

- 1: Ermittle die H-Darstellung und die Ecken von $V_I(\bar{p})$
 - 2: Berechne zu jeder Ecke y^l die jeweilige Stützhyperebene H^l
 - 3: Stelle das lineare Optimierungsproblem (4.4) auf und löse es
 - 4: Die Strategie ist $\begin{cases} \text{minimal,} & \text{Optimalwert} = 0 \\ \text{nicht minimal,} & \text{sonst} \end{cases}$
-

Algorithmus 3 Pseudocode Berechnung der Stützhyperebene für eine bestimmte Ecke y^l eines in H-Darstellung gegebenen Polyeders $V_I(\bar{p})$

- 1: Setze die Ecke in die H-Darstellung ein
 - 2: Bestimme diejenigen Ungleichungen in denen Gleichheit vorliegt
 - 3: Summiere die entsprechenden Ungleichungen auf
 - 4: Die Summe repräsentiert die Parameter der Stützhyperebenen
-

Algorithmus 4 Pseudocode Test eines Strategiepaars auf ein Shapley-Gleichgewicht

- 1: Berechne $v(\bar{p}, \bar{q})$
 - 2: Ermittle $V_I(\bar{p})$ und $V_{II}(\bar{q})$ sowie deren H-Darstellungen
 - 3: Bestimme jeweils die Ungleichungen der H-Darstellungen in denen im Punkt $v(\bar{p}, \bar{q})$ Gleichheit vorliegt
 - 4: Summiere jeweils die entsprechenden Matrixspalten auf
 - 5: Strategiepaar ist $\begin{cases} \text{ein Shapley-Gleichgewicht,} & \text{Summen} \in \text{int } \mathbb{R}_+^K \text{ bzw. } \text{int } \mathbb{R}_-^K \\ \text{kein Shapley-Gleichgewicht,} & \text{sonst} \end{cases}$
-

Bei dem Test auf ein Shapley-Gleichgewicht werden erneut die H-Darstellungen der Polyeder $V_I(\bar{p})$ und $V_{II}(\bar{q})$ verwendet, um Stützhyperebenen zu ermitteln.

$$V_I(\bar{p}) = \{y \in \mathbb{R}^K \mid B \cdot y \leq b\}$$

Die Bestimmung der entsprechenden Hyperebenen erfolgt, indem die Ungleichungsrestriktionen der H-Darstellung, die im Punkt $v(\bar{p}, \bar{q})$ mit Gleichheit erfüllt sind, identifiziert werden. Anschließend werden die betroffenen Zeilen der Matrix B des Polyeders $V_I(\bar{p})$ aufsummiert. Die Summe liegt im $\text{int } \mathbb{R}_+^K$ genau dann, wenn die resultierende Auszahlung des Strategiepaars (\bar{p}, \bar{q}) Teil der maximal erwarteten Auszahlung für den Spieler I unter Verwendung der Strategie \bar{p} ist. Ob $v(\bar{p}, \bar{q})$ in $\text{Min } v_{II}(\bar{q})$ liegt, lässt sich auf analoge Weise nachweisen. Das heißt, dass überprüft wird, ob die Summe im $\text{int } \mathbb{R}_-^K$ liegt. Ist $v(\bar{p}, \bar{q})$ in beiden Mengen enthalten ($\text{Max } v_I(\bar{p}), \text{Min } v_{II}(\bar{q})$), liegt ein Shapley-Gleichgewicht vor [vgl. 7, S. 22]. In Algorithmus 4 wird ein Strategiepaar auf ein Shapley-Gleichgewicht getestet.

Soll das Strategiepaar (\bar{p}, \bar{q}) auf ein starkes Shapley-Gleichgewicht geprüft werden muss überprüft werden, ob

$$V_I(\bar{p}) \cap V_{II}(\bar{q}) \subseteq \text{Max } v_I(\bar{p}) \cap \text{Min } v_{II}(\bar{q}). \quad (4.5)$$

(4.5) ist äquivalent dazu, dass für alle $\varepsilon > 0$ und alle $k \in \{1, \dots, K\}$

$$V_I(\bar{p}) \cap (V_{II}(\bar{q}) + \varepsilon \cdot e^k) = \emptyset. \quad (4.6)$$

e^k repräsentiert hierbei den k -ten Einheitsvektor des \mathbb{R}^K . Seien die H-Darstellungen der Polyeder $V_I(\bar{p})$ und $V_{II}(\bar{q})$ in der Form

$$\begin{aligned} V_I(\bar{p}) &= \{y \in \mathbb{R}^K \mid A \cdot y \geq a\} \\ V_{II}(\bar{q}) &= \{y \in \mathbb{R}^K \mid B \cdot y \geq b\} \end{aligned}$$

gegeben. Dann ist der Optimalwert des linearen Optimierungsproblems (4.7) gleich null genau dann, wenn die Gleichung (4.6) erfüllt ist [vgl. 7, S. 22f].

$$\begin{aligned} \max_{y,t} \quad & e^T \cdot t \\ \text{bezüglich:} \quad & A \cdot y \geq a \\ & B \cdot (y - t) \geq b \\ & t \geq 0 \end{aligned} \tag{4.7}$$

Anhand des linearen Problems (4.7) wird in Algorithmus 5 geprüft, ob es sich bei einem gegebenen Strategiepaar um ein starkes Shapley-Gleichgewicht handelt.

Algorithmus 5 Pseudocode Test eines Strategiepaares auf ein starkes Shapley-Gleichgewicht

- 1: Ermittle $V_I(\bar{p})$ und $V_{II}(\bar{q})$ sowie deren H-Darstellungen
 - 2: Stelle das lineare Optimierungsproblem (4.7) auf und löse es
 - 3: Das Strategiepaar ist $\begin{cases} \text{ein starkes Shapley-Gleichgewicht,} & \text{Optimalwert} = 0 \\ \text{kein starkes Shapley-Gleichgewicht,} & \text{sonst} \end{cases}$
-

Die Implementierungen der vorgestellten Algorithmen und alle Codedateien, die die Abbildungen dieser Arbeit erzeugen, befinden sich auf <https://github.com/SarahStudent/Bachelorarbeit>. Dort sind außerdem sämtliche Abbildungen dieser Arbeit abrufbar.

4.1 Implementierungsdetails

Für die Implementierung der Algorithmen des vorherigen Abschnittes wird *bensolve tools* [2, 3, 4, 11] verwendet. Die kostenlose Software umfasst Anweisungen zur Berechnung konvexer Polyeder und polyedrischer konvexer Funktionen. Sie enthält auch verschiedene Solver für Optimierungsprobleme. Dabei werden der VLP-Solver *bensolve* [12, 10] und das GNU Linear Programming Kit (GLPK) verwendet [vgl. 3]. *bensolve* [12, 10] ist eine Open-Source-Implementierung des Benson-Algorithmus und berechnet Lösungen von linearen Vektoroptimierungsproblemen [vgl. 10].

Lineare Optimierungsprobleme, wie beispielsweise (4.4) und (4.7), können durch *lpsolve* der *bensolve tools* [2, 3, 4, 11] gelöst werden. *lpsolve* errechnet die Lösung von linearen Optimierungsproblemen der Form

$$\min c^T x \quad \text{bzgl.} \quad x \in S.$$

Es müssen demnach eine Optimierungsrichtung („*min*“ oder „*max*“), ein Vektor c , der die Zielfunktion repräsentiert und ein Polyeder S in P-Darstellung, der den zulässigen Bereich des Optimierungsproblems darstellt, an den Solver übergeben werden. Das Ergebnis besteht aus dem Optimalwert, den primalen und dualen Optimallösungen sowie dem Status des Problems [vgl. 3, Manual S. 48]. Der Solver wird in den entwickelten Implementierungen, die sich teilweise in Abschnitt 7 befinden, mehrfach verwendet.

Soll ein Strategiepaar (\bar{p}, \bar{q}) auf Optimalität und (starke) Shapley-Gleichgewichte getestet werden, bietet es sich an, eine Implementierung zu verwenden, die alle bisherigen Algorithmen kombiniert. Der Aufwand und somit auch die Laufzeit können reduziert werden, indem die Polyeder $V_I(\bar{p})$ und $V_{II}(\bar{q})$, sowie deren H-Darstellungen einmalig berechnet werden und dann an die jeweiligen Algorithmen direkt übergeben werden [vgl. 7, S. 22].

Für weitere Implementierungsdetails wird auf <https://github.com/SarahStudent/Bachelorarbeit> verwiesen.

5 Numerische Ergebnisse

Die Algorithmen der vorigen Abschnitte wurden in MATLAB Version 9.13.0 implementiert. Außerdem wurde *bensolve tools* Version 1.3 [2, 3, 4, 11] verwendet. Jede Berechnung wurde auf einem Computer mit Intel®Core™ i5 CPU mit 1.6GHz und 8GB RAM ausgeführt.

Um eine endliche Repräsentation der Menge an optimalen Strategien für Spieler I zu ermitteln, wird die Menge der zulässigen Strategien P mit der Schrittweite $t > 0$ diskretisiert:

$$\bar{P} = P \cap \{x \in \mathbb{R}^K \mid \forall k = \{1, \dots, K\} \exists z_k \in \mathbb{Z} : x_k = z_k \cdot t\}.$$

Analog wird die Menge Q an zulässigen Strategien der Spielerin II ebenfalls mit einer Schrittweite $t > 0$ diskretisiert [vgl. 7, S. 23]. Die Spiele und die Schrittweiten in den nachfolgenden Beispielen 5.1 und 5.2 stammen jeweils aus [7, S.23f Beispiel 5.1 und Beispiel 5.2].

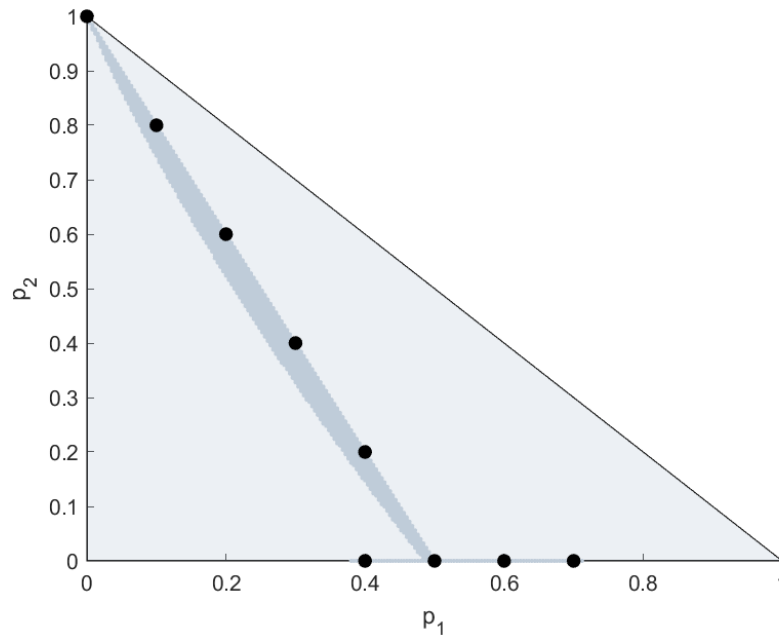


Abbildung 5: Abgebildet sind für den Spieler I die minimalen Strategien für das Spiel aus Beispiel 5.1. Sie sind als dunkelgraue Bereich dargestellt. Der hellgraue Bereich repräsentiert die ersten beiden Komponenten aller zulässigen Strategien. Die dritte Komponente lässt sich aus den anderen Komponenten berechnen. Die schwarzen Punkte stellen ausgewählte optimale Strategien dar, deren Auszahlungen in Abbildung 7 abgebildet sind [vgl. 7, Abbildung 5.1 S. 24].

Beispiel 5.1. Sei das Spiel

$$A = \begin{pmatrix} \begin{pmatrix} 5 \\ 0 \end{pmatrix} & \begin{pmatrix} -1 \\ -5 \end{pmatrix} & \begin{pmatrix} 4 \\ -4 \end{pmatrix} \\ \begin{pmatrix} 2 \\ -2 \end{pmatrix} & \begin{pmatrix} 2 \\ -7 \end{pmatrix} & \begin{pmatrix} 2 \\ 2 \end{pmatrix} \\ \begin{pmatrix} 0 \\ -6 \end{pmatrix} & \begin{pmatrix} 6 \\ -2 \end{pmatrix} & \begin{pmatrix} -2 \\ 4 \end{pmatrix} \end{pmatrix}$$

gegeben [vgl. 7, S. 23f]. Mit der Schrittweite $\frac{1}{10}$ werden neun minimale Strategien für Spieler I gefunden und für Spielerin II ergeben sich mit der Schrittweite $\frac{1}{5}$ sechs maximale Strategien. Sie sind in den Abbildungen 5 und 6 als schwarze Punkte abgebildet. Unter den resultierenden 54 optimalen Strategiepaaren befinden sich elf Shapley-Gleichgewichte von denen zwei stark sind, siehe Tabelle 2. Die optimalen Strategien, die sich mit der Schrittweite $\frac{1}{250}$ berechnet lassen, befinden sich in Abbildung 5 und 6. Die zugehörigen Auszahlungen und deren Verhältnis zu den POSS Auszahlungen sind in Abbildung 7 dargestellt.

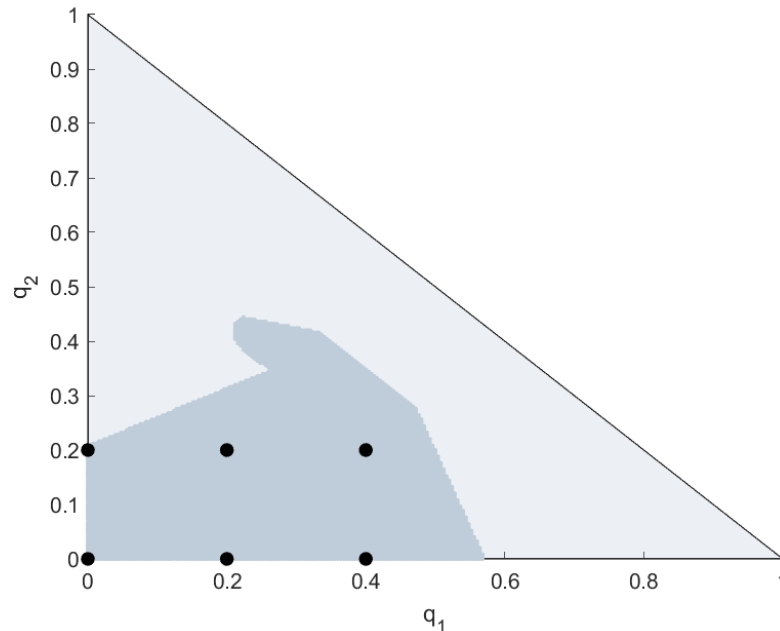


Abbildung 6: Die Abbildung ist analog zu Abbildung 5 aufgebaut und bildet die maximalen Strategien innerhalb des Spiels aus Beispiel 5.1 für die Spielerin II ab. Die schwarzen Punkte stellen auch hier ausgewählte optimale Strategien dar, deren Auszahlungen in Abbildung 7 abgebildet sind [vgl. 7, Abbildung 5.1 S. 24].

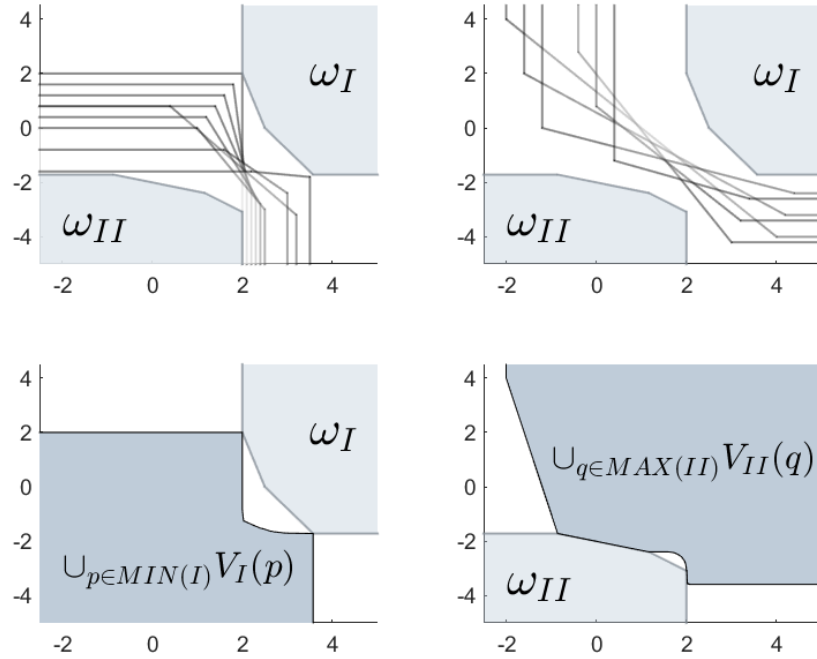


Abbildung 7: Abgebildet sind verschiedene Auszahlungen aus Beispiel 5.1. ω_I und ω_{II} sind die Auszahlungen der POSS (siehe Unterabschnitt 3.3). Die oberen Abbildungen zeigen die Grenzen der Mengen $V_I(p)$ (links) und $V_{II}(q)$ (rechts) von ausgewählten optimalen Strategien. Die entsprechenden Strategien sind aus den Abbildungen 5 und 6 entnehmbar. In den unteren Abbildungen ist eine Approximation der Vereinigungen der Auszahlungen, die durch alle minimalen und maximalen Strategien erzeugt werden, dargestellt. Es lässt sich erkennen, dass der Ansatz nach Hamel und Löhne bessere Ergebnisse erzielt als POSS [vgl. 7, Abbildung 5.2 S. 25].

p^T	q^T	Typ	p^T	q^T	Typ
$(\frac{2}{5}, 0, \frac{3}{5})$	$(0, 0, 1)$	stark	$(\frac{3}{5}, 0, \frac{2}{5})$	$(\frac{1}{5}, 0, \frac{4}{5})$	nicht stark
$(\frac{1}{2}, 0, \frac{1}{2})$	$(0, 0, 1)$	stark	$(\frac{7}{10}, 0, \frac{3}{10})$	$(\frac{1}{5}, 0, \frac{4}{5})$	nicht stark
$(\frac{3}{5}, 0, \frac{2}{5})$	$(0, 0, 1)$	nicht stark	$(\frac{1}{2}, 0, \frac{1}{2})$	$(\frac{2}{5}, 0, \frac{3}{5})$	nicht stark
$(\frac{7}{10}, 0, \frac{3}{10})$	$(0, 0, 1)$	nicht stark	$(\frac{3}{5}, 0, \frac{2}{5})$	$(\frac{2}{5}, 0, \frac{3}{5})$	nicht stark
$(\frac{2}{5}, 0, \frac{3}{5})$	$(\frac{1}{5}, 0, \frac{4}{5})$	nicht stark	$(\frac{7}{10}, 0, \frac{3}{10})$	$(\frac{2}{5}, 0, \frac{3}{5})$	nicht stark
$(\frac{1}{2}, 0, \frac{1}{2})$	$(\frac{1}{5}, 0, \frac{4}{5})$	nicht stark			

Tabelle 2: Shapley-Mengen-Gleichgewichtspunkte des Beispiels 5.1 [vgl. 7, Tabelle 5.1 S. 23].

K=2	n			
	m	2	3	4
2	0.31			
3	1.77	3.36		
4	22.91	28.13	37.91	

K=3	n			
	m	2	3	4
2	0.37			
3	2.19	4.04		
4	28.01	33.6	49.76	

K=4	n			
	m	2	3	4
2	0.45			
3	2.75	5.11		
4	35.68	39.57	61.72	

K=5	n			
	m	2	3	4
2	0.57			
3	3.34	6.84		
4	46.96	53.27	93.02	

Tabelle 3: Durchschnittliche Laufzeiten in Sekunden des Beispiels 5.2 bei fünf Durchläufen und einer Schrittweite $t = \frac{1}{20}$ [vgl. 7, Tabelle 5.2 S. 24].

Beispiel 5.2. Seien die einzelnen Komponenten der Vektoren $a_{ij} \in \mathbb{R}^K$ einer Matrix A zufällige Ganzzahlen zwischen -10 und 10 . Es werden Spiele mit der Auszahlungsmatrix A unter Verwendung der Schrittweite $t = \frac{1}{20}$ gelöst, bei denen die Matrixgröße (m,n) sowie die Dimension der Auszahlungen (K) variieren. Hierbei sind $m \in \{2, 3, 4\}$, $n \in \{2, 3, 4\}$ und $K \in \{2, 3, 4, 5\}$. Für jede Wahl von m, n, K wird das zufällige Spiel fünf mal gelöst [vgl. 7, S.23f]. Die durchschnittlichen Laufzeiten der Berechnung optimaler Strategien befinden sich in Tabelle 3.

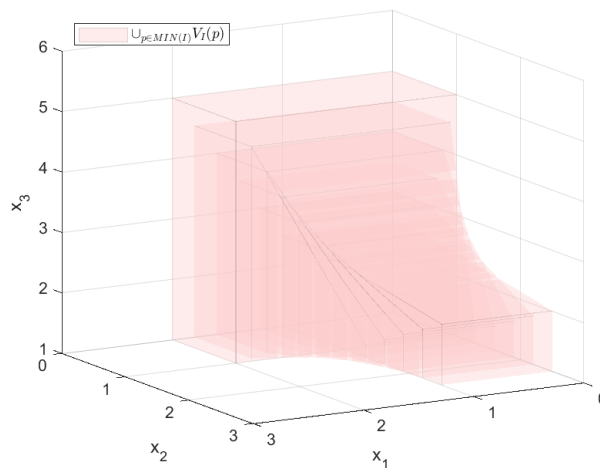


Abbildung 8: Für das Spiel in Beispiel 5.3 sind bestimmte Mengen $V_I(p)$ abgebildet. Sie werden durch die minimalen Strategien der Schrittweite $t = \frac{1}{10}$ erzeugt und dienen dazu, einen Eindruck von der Vereinigung der gesamten $V_I(p)$ der minimalen Strategien zu schaffen.

Beispiel 5.3. Für das Spiel

$$A = \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 4 \\ 3 \end{pmatrix} \\ \begin{pmatrix} 2 \\ 1 \\ 5 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \end{pmatrix}$$

besteht die Menge der optimalen Strategien für die Schrittweite $t = \frac{1}{200}$ aus

$$\text{MIN}(I) = \{p \in P \mid 0 \leq p_1 \leq 0.57\} \quad \text{und} \quad \text{MAX}(II) = \{q \in Q \mid 0 \leq q_1 \leq 0.745\}.$$

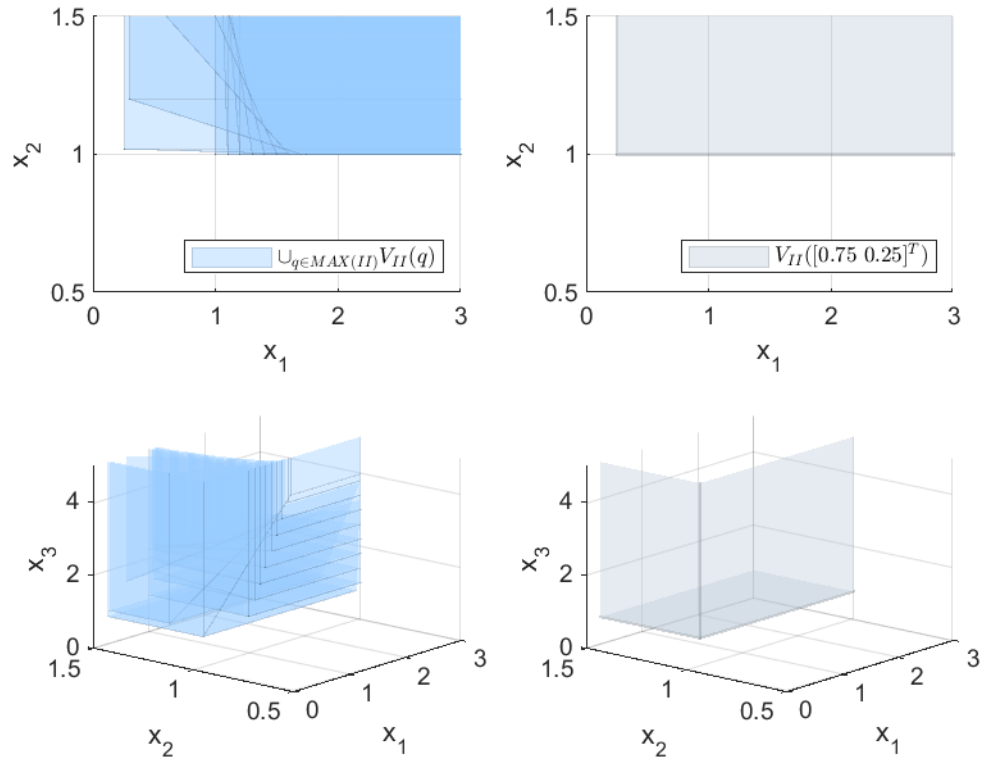


Abbildung 9: Abgebildet sind mehrere $V_{II}(q)$ von verschiedenen Strategien der Spielerin II für das Spiel in Beispiel 5.3. In blau sind in den linken beiden Graphiken aus zwei verschiedenen Perspektiven die $V_{II}(q)$ abgebildet, die von den maximalen Strategien der Schrittweite $t = \frac{1}{10}$ und der Strategie $q = (0.745, 0.255)^T$ erzeugt werden. Mit den selben Perspektiven befindet sich rechts in grau $V_{II}(q)$ mit der Strategie $q = (0.75, 0.25)^T$. Es lässt sich erkennen, dass die blauen Polyeder den grauen Polyeder approximieren.

6 Diskussion

Im Rahmen der vorliegenden Arbeit wurde eine Implementierung des Algorithmus aus [7] entwickelt, die die *bensolve tools* [2, 3, 4, 11] verwendet. Mit der Implementierung sollen die optimalen Strategien eines Matrixspiels bestimmt werden. Die Ergebnisse des vorherigen Abschnitts 5 zeigen, dass die Implementierung die optimalen Strategien allgemeiner Spiele korrekt ermitteln kann. Allerdings ist eine signifikant schlechtere Laufzeit im Vergleich zu der ursprünglichen Implementierung von Hamel und Löhne mit steigender Größe der Matrix des Spiels zu beobachten [vgl. 7, S. 24 Tabelle 5.2].

K=2				K=3			
m	n			m	n		
	2	3	4		2	3	4
2	0.11			2	0.07		
3	1.17	1.66		3	0.59	0.64	
4	19.01	20.33	23.91	4	20.41	23.6	19.76

K=4				K=5			
m	n			m	n		
	2	3	4		2	3	4
2	0.05			2	0.17		
3	0.75	0.41		3	1.34	1.14	
4	21.28	19.37	16.72	4	30.86	23.67	28.02

Tabelle 4: Differenz der durchschnittlichen Laufzeiten in Sekunden des Beispiels 5.2 bei fünf Durchläufen und einer Schrittweite $t = \frac{1}{20}$ zwischen der entwickelten und ursprünglichen Implementierung. Die Differenzen ergeben sich aus den Werten der Tabelle 3 und Tabelle 5.2 aus [7]. Sie geben die Anzahl der Sekunden an, die die entwickelte Implementierung länger braucht, um die optimalen Strategien zu berechnen.

Erwartungsgemäß kann die Annahme, dass eine Implementierung des Optimalitätstests unter Verwendung der *bensolve tools* [2, 3, 4, 11] möglich ist, bestätigt werden. Das lässt sich damit erklären, dass eine funktionierende Implementierung des Algorithmus unter Verwendung des Solvers *bensolve* [12, 10] bereits existiert [vgl. 7] und der Solver von *bensolve tools* [2, 3, 4, 11] intern genutzt wird [vgl. 3]. Des Weiteren war zu erwarten, dass sich die Laufzeit der erarbeiteten Implementierung erhöht. Das ergibt sich aus demselben Grund, wieso eine erfolgreiche Implementierung überhaupt zu erwarten ist: *bensolve tools* [2, 3, 4, 11] verwendet *bensolve* [12, 10]. Dadurch entsteht jedes Mal ein zusätzlicher Aufwand, wenn Funktionen in *bensolve tools* [2, 3, 4, 11] aufgerufen werden, die aus *bensolve* [12, 10] stammen. Ein größerer

Aufwand hat eine höhere Laufzeit zur Folge. Ziel dieser Arbeit ist die Entwicklung einer übersichtlichen und sofort nutzbaren Implementierung mittels *bensolve tools* [2, 3, 4, 11]. Sie soll ausführlich dokumentiert und öffentlich zugänglich sein. Zum einen soll sie eine Ergänzung zu dem Lösungsansatz von Hamel und Löhne in [8] darstellen und zum anderen sollen dadurch weiterführende Forschungen bezüglich der optimalen Strategien erleichtert werden. Das Ziel der Arbeit besteht nicht darin, eine Implementierung mit minimaler Laufzeit zu entwickeln. Allerdings ergibt sich daraus die Möglichkeit, die hier ausgearbeitete Implementierung hinsichtlich Laufzeit und Effizienz zu optimieren.

7 Zusammenfassung

Hamel und Löhne stellen zur Lösung von Spielmatrizen mit mehrdimensionaler Auszahlung einen Ansatz basierend auf Mengenrelationen bereit [vgl. 8, S. 372]. Dabei bilden optimale Strategien die Lösungen des Spiels [vgl. 8, S. 396]. Die ursprüngliche Implementierung von Hamel und Löhne des Algorithmus zur Bestimmung der optimalen Strategien verwendet *bensolve* [12, 10] [vgl. 7, S. 23]. Ebenfalls für die Implementierung geeignet, ist die Software *bensolve tools* [2, 3, 4, 11]. Die damit entwickelte Implementierung erzielt korrekte Ergebnisse, weist jedoch schlechtere Laufzeiten als die ursprüngliche Implementierung auf.

Literaturverzeichnis

- [1] M. Archetti, D. A. Ferraro und G. Christofori. “Heterogeneity for IGF-II production maintained by public goods dynamics in neuroendocrine pancreatic cancer”. In: *Proceedings of the National Academy of Sciences* 112.6 (2015), S. 1833–1838.
- [2] D. Ciripoi, A. Löhne und B. Weißing. “A vector linear programming approach for certain global optimization problems”. In: *Journal of Global Optimization* 72.2 (2018), S. 347–372.
- [3] D. Ciripoi, A. Löhne und B. Weißing. *bensolve tools*. Version 1.3. Gnu Octave/ Matlab toolbox for calculus of convex polyhedra, calculus of polyhedral convex functions, global optimization, vector linear programming, Aufgerufen: 25.3.2023. 2019. URL: <http://tools.bensolve.org/>.
- [4] D. Ciripoi, A. Löhne und B. Weißing. “Calculus of convex polyhedra and polyhedral convex functions by utilizing a multiple objective linear programming solver”. In: *Optimization* 68.10 (2019), S. 2039–2054.
- [5] Dudenredaktion, Hrsg. *Duden Die deutsche Rechtschreibung*. 27. Aufl. Dudenverlag, 2017.
- [6] M. Felegyhazi und J. Hubaux. “Game Theory in Wireless Networks: A Tutorial”. In: *ACM Comput. Surveys* (Jan. 2006).
- [7] A. Hamel und A. Löhne. “A set optimization approach to zero-sum matrix games with multi-dimensional payoffs”. Preprint, unveröffentlicht. 2017.
- [8] A. Hamel und A. Löhne. “A set optimization approach to zero-sum matrix games with multi-dimensional payoffs”. In: *Mathematical Methods of Operations Research* 88.3 (2018), S. 369–397.
- [9] H. Küpper. *Wörterbuch der deutschen Umgangssprache*. 1. Aufl., 4. Nachdr. PONS. Klett, 1990, S. 40.
- [10] A. Löhne und B. Weißing. *Bensolve*. Version 2.1.0. A free vector linear program solver, Aufgerufen: 25.3.2023. 2017. URL: <http://www.bensolve.org/>.
- [11] A. Löhne und B. Weißing. “Equivalence between polyhedral projection, multiple objective linear programming and vector linear programming”. In: *Mathematical Methods of Operations Research* 84.2 (2016), S. 411–426.
- [12] A. Löhne und B. Weißing. “The vector linear program solver Bensolve – notes on theoretical background”. In: *European Journal of Operational Research* 260.3 (2017), S. 807–813.

- [13] M. Maschler, E. Solan und S. Zamir. *Game Theory*. Cambridge University Press, 2013.
- [14] C. H. Papadimitriou. *Algorithmic Game Theory*. Hrsg. von N.Nisan u. a. Cambridge University Press, 2007.
- [15] A. Rubinstein und M. Osborne. *A Course in Game Theory*. MIT Press, 1994.
- [16] F. Schiller. "Über die ästhetische Erziehung des Menschen". In: *Die Horen* 2 (1795), S. 51–94.
- [17] L. S. Shapley. "Equilibrium points in games with vector payoffs". In: *Naval Research Logistics Quarterly* 6.1 (1959), S. 57–61.
- [18] J. Von Neumann und O. Morgenstern. *Spieltheorie und wirtschaftliches Verhalten*. Übers. von M. Lepping. Physica-Verlag, 1961.
- [19] J. Von Neumann und O. Morgenstern. *Theory of Games and Economic Behavior*. 2. Aufl. Princeton University Press, 1947.
- [20] H. Wiese. *Entscheidungs- und Spieltheorie*. 1. Aufl. Springer-Lehrbuch. Springer Berlin, Heidelberg, 2002.

Verzeichnis der Abbildungen

1	$V_I(\bar{p})$ und $V_{II}(\bar{q})$ für die Strategien und das Spiel aus Beispiel 2.1 . . .	12
2	Grafische Ermittlung der minimalen Strategien für das Spiel aus Beispiel 2.1	13
3	Die zu prüfenden Größen für ein Shapley-Gleichgewicht für zwei Strategien aus Beispiel 3.6	16
4	Die zu prüfenden Größen für ein Shapley-Gleichgewicht für zwei Strategien aus Beispiel 3.6	17
5	Minimale Strategien für das Spiel in Beispiel 5.1	25
6	Maximale Strategien für das Spiel in Beispiel 5.1	26
7	Auszahlungen minimaler und maximaler Strategien des Beispiels 5.1 und dessen Auszahlungen der POSS	27
8	$V_I(p)$ für $p \in \text{MIN}(I)$ mit $t = \frac{1}{10}$ aus Beispiel 5.3	28
9	$V_{II}(q)$ für $q \in \text{MAX}(II)$ mit $t = \frac{1}{10}$ aus Beispiel 5.3 in Vergleich zu $V_{II}((0.75, 0.25))^T$	29

Verzeichnis der Tabellen

1	Matching Pennies	7
2	Shapley-Mengen-Gleichgewichtspunkte des Spiels in Beispiel 5.1 . . .	27
3	Durchschnittliche Laufzeiten in Sekunden bei fünf Durchläufen für Beispiel 5.2	28
4	Differenz der durchschnittlichen Laufzeiten in Sekunden zwischen der entwickelten und ursprünglichen Implementierung	30

Verzeichnis der Algorithmen

1	Pseudocode Berechnung von $V_I(p)$	12
2	Pseudocode Optimalitätstest Strategie \bar{p}	21
3	Pseudocode Berechnung der Stützhyperebene	22
4	Pseudocode Test eines Strategiepaares auf ein Shapley-Gleichgewicht	22
5	Pseudocode Test eines Strategiepaares auf ein starkes Shapley- Gleichgewicht	23

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt, alle benutzten Quellen und Hilfsmittel angegeben und Zitate als solche kenntlich gemacht habe.

Es bestehen keine Einwände die vorliegende Bachelorarbeit für die öffentliche Benutzung im Universitätsarchiv zur Verfügung zu stellen.

Ich versichere ferner, dass ich die Arbeit weder für eine Prüfung an einer weiteren Hochschule noch für eine staatliche Prüfung eingereicht habe.

Jena, den 22.03.2023

Anlagen

```
1  g1 = {[0;0], [3;2]; [2;0], [1;2]};
2  p = [0;1];
3  V1 = calculateV1(p,g1);
4  V1 = eval(V1);
5  B1 = hrep(V1).B;
6  b1 = hrep(V1).b;
7  ecken1 = vrep(V1).V;
8  [p_min] = optimizetest(g1,B1,b1,ecken1,10^-6);
9  % ist p_min == 0, liegt eine minimale Strategie für den Spieler vor, sonst nicht
10
11 function [V1] = calculateV1(p,g)
12   K = length(g{1,1});
13   [m,n] = size(g);
14   V = zeros([K n]);
15   for j = 1:n
16       sum = 0;
```

```

17     for i = 1:m
18         sum = sum + p(i) * g{i,j};
19     end
20     V(:,j) = sum;
21 end
22 rep1.V = V;
23 V1 = polyh(rep1,'v');
24 V1 = V1 - cone(K);
25 end
26
27 function [c,gamma] = hyperplane(e,B,b,epsilon)
28     c = zeros(1,size(B,2));
29     gamma = 0;
30     f = B*e-b;
31     for i = 1:size(f)
32         if abs(f(i)) <= epsilon % z.B. 10^-6
33             c = c + B(i,:);
34             gamma = gamma + b(i);
35         end
36     end
37     c = -1*c;
38     gamma = -1*gamma;
39 end
40
41 function [optval,sol_p,sol_d,status] = optimizetest(g,B,b,ecken,epsilon)
42     H = -1*B; % geht aus dem manual für bensolvetools hervor
43     h = -1*b;
44     r = size(ecken,2);
45     [m,n] = size(g);
46     % initialisierung der parameter
47     repB = zeros(n*(size(H*g{1,1},1)+r)+1,m+r);
48     repa = zeros(n*(size(H*g{1,1},1)+r)+1,1);
49     repb = -log(0) * ones(n*(size(H*g{1,1},1)+r)+1,1);
50     repl = zeros(1,m+r);
51     d = zeros(1,m+r);
52
53     %jetzt überall schrittweise werte auffüllen

```



```

54  for i = 1:r
55      repl(m+i) = log(0);
56      d(m+i) = 1;
57  end
58
59  k1 = 0; % werte der H darstellungen werden eingelesen
60  for j = 1:n
61      for i = 1:m
62          x = H*g{i,j};
63          for k = 1:size(x,1)
64              repB(k+k1,i) = x(k);
65              repa(k+k1,1) = h(k);
66          end
67      end
68      k1 = k1+size(x,1);
69  end
70
71  % ab hier werden die werte der hyperebenen-ungleichung eingesetzt
72  k1 = n*size(H*g{1,1},1)+1;
73  k2 = 1;
74  for id = eye(r)
75      [c,gamma] = hyperplane(ecken*id,B,b,epsilon);
76      for j = 1:n
77          for i = 1:m
78              repB(k1,i) = c*g{i,j};
79              repa(k1,1) = gamma;
80          end
81          repB(k1,m+k2) = -1;
82          k1 = k1+1;
83      end
84      k2 = k2+1;
85  end
86  for i = 1:m % letzte Zeile
87      repB(k1,i) = 1;
88  end
89  repa(k1,1) = 1;
90  repb(k1,1) = 1;

```

```
91 clear rep;  
92 rep.B = repB;  
93 rep.a = repa;  
94 rep.b = repb;  
95 rep.l = (repl)';  
96 rep.M = eye(m+r);  
97 S = polyh(rep);  
98 d = d';  
99 [optval,sol_p,sol_d,status] = lpsolve(d,S,'max');  
100 end
```
