



FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Ein Algorithmus für konvexe Mengenoptimierungsprobleme

Masterarbeit zur Erlangung des akademischen Grades
Master of Science (M. 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. Daniel Dörfler

Jena, 28.01.26

Kurzfassung

Im Rahmen dieser Arbeit wird eine Methode zur Lösung kompakt-konvexer Mengenoptimierungsprobleme vorgestellt. Bei diesen Optimierungsproblemen wird eine mengenwertige Abbildung $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$, deren Graph kompakt und konvex ist, bezüglich der Mengeninklusion \subseteq maximiert. Der entwickelte Lösungsalgorithmus wird ohne weitere Annahmen als korrekt bewiesen. Die Endlichkeit des Algorithmus gilt unter der zusätzlichen Voraussetzung, dass der Ursprung im Inneren des Graphen liegt. Zwei Beispiele werden im Zuge einer Implementierung gelöst.

Inhaltsverzeichnis

Symbolverzeichnis	iv
1 Einleitung	1
2 Mathematische Grundlagen	3
2.1 Notationen	3
2.2 Hausdorffabstand	5
2.3 Mengenoptimierung	5
3 Existenz von Optimierern	7
4 Algorithmus zur Berechnung von ϵ-Optimierern	10
4.1 Korrektheit	13
4.2 Endlichkeit	14
5 Beispiele	18
5.1 Wachsender Kreis	18
5.2 Portfoliooptimierung	19
5.3 Implementierungsdetails	22
6 Diskussion	24
7 Fazit	25
Literaturverzeichnis	28
Verzeichnis der Algorithmen	29
Abbildungsverzeichnis	30
Eigenständigkeitserklärung	31

Symbolverzeichnis

\mathbb{R}	Menge der reellen Zahlen
\mathbb{R}^n	n -dimensionaler, euklidischer Raum
$\mathbb{R}^{n \times m}$	Menge der reellen $n \times m$ Matrizen
e_j	Vektor mit 1 an der j -ten Stelle und sonst 0
e	Vektor, dessen Komponenten alle gleich 1 sind
$(\cdot)^T$	transponiert
$x^T y$	Skalarprodukt von $x \in \mathbb{R}^n$ und $y \in \mathbb{R}^n$
$\text{dom } F$	Definitionsbereich der Funktion F
$\text{gr } F$	Graph der Funktion F
$\sigma_C(x)$	Stützfunktion der Menge C in x
$\text{conv } C$	konvexe Hülle von C
$\text{cone } C$	konische Hülle von C
$\text{aff } C$	affine Hülle von C
$\text{int } C$	Innere der Menge C
$\text{cl } C$	Abschluss der Menge C
$\text{bd } C$	Rand der Menge C
$d(y, C)$	euklidischer Abstand zwischen y und C
$\ \cdot\ $	euklidische Norm
$B_r(x)$	euklidischer Ball mit Radius r und Mittelpunkt x
B	euklidischer Einheitsball: Radius 1, Mittelpunkt 0
$e[C_1, C_2]$	Exzess von C_1 zu C_2
$d_H(C_1, C_2)$	Hausdorffabstand zwischen C_1 und C_2

1 Einleitung

Die Mengenoptimierung stellt ein Teilgebiet der angewandten Mathematik dar, bei dem Optimierungsprobleme mit mengenwertigen Zielfunktionen und/oder Restriktionen untersucht werden. Da die mengenwertigen Funktionen die skalaren Funktionen mit umfassen, handelt es sich bei der Mengenoptimierung sowohl um eine Erweiterung als auch eine Verallgemeinerung der skalaren und multi-kriteriellen Optimierung [vgl. 22, S. 1]. Demnach verfügt die Mengenoptimierung über ein vielfältiges Anwendungsfeld [vgl. 15, S. 119]. Als praktische Anwendungsbereiche lassen sich beispielsweise die Wohlfahrtsökonomie [vgl. 4], die Finanzmathematik [vgl. 13, S. vi] und die Bildverarbeitung [vgl. 22, S. 1] nennen. Darüber hinaus können Optimierungsprobleme anderer Forschungsgebiete oftmals als mengenwertige Probleme formuliert und dementsprechend untersucht werden [vgl. 22, S. 1]. Beispiele hierfür sind Probleme der stochastischen Optimierung, der unscharfen Optimierung und der Streuungstheorie [vgl. 16, S. 37].

Aus der Literatur gehen die beiden nachfolgenden, meistgenutzten Ansätze zur Lösung allgemeiner Mengenoptimierungsprobleme hervor: ein mengenwertiger Ansatz, bei dem alle oder einige Optimierer berechnet werden, ohne dass das Infimum¹ oder Supremum² angenommen werden muss, und ein vollständiger Verbundsansatz, wo Optimierer gesucht werden, die das Infimum oder Supremum annehmen [vgl. 28, S. 3]. Ein Lösungsansatz, der auf dem letztgenannten Ansatz beruht, ist die Lösungsmethode nach Löhne [29]. Der von Löhne entwickelte Algorithmus bestimmt innerhalb endlicher Zeit einen Optimierer für polyedrisch-konvexe Probleme, eine spezielle Unterklasse von Mengenoptimierungsproblemen [vgl. 29, Kapitel 3].

Allerdings stellen polyedrisch-konvexe Probleme lediglich einen Teil aller Mengenoptimierungsprobleme dar. Idealerweise sollte der Lösungsansatz über diesen speziellen Problemtyp hinaus auch auf allgemeinere Problemklassen anwendbar sein, wie beispielsweise auf konvexe Mengenoptimierungsprobleme. Die Entwicklung einer Lösungsmethode für kompakt-konvexe Probleme kann dabei einen ersten Schritt in diese Richtung darstellen, von dem aus allgemeinere konvexe Probleme weiter untersucht werden können.

In dieser Arbeit soll, angelehnt an das Lösungskonzept der Optimierer nach Löhne [28, 29], eine Lösungsmethode für kompakt-konvexe Mengenoptimierungsprobleme entwickelt werden. Der dabei entstehende Lösungsalgorithmus soll sowohl korrekt als auch endlich sein. Zudem soll ein praxisnahes Beispiel durch eine Implemen-

¹Das Infimum bezeichnet die untere Grenze einer beschränkten Menge [vgl. 8, S. 578].

²Das Supremum bezeichnet die obere Grenze einer beschränkten Menge [vgl. 8, S. 1079].

tierung des Algorithmus gelöst werden. Für die Entwicklung der Lösungsmethode wird sich an dem Kernelgorithmus zur Lösung polyedrisch-konvexer Mengenoptimierungsprobleme nach Löhne [29, Algorithmus 1] orientiert, da dieser bereits für bestimmte konvexe Probleme eine korrekte Lösung in endlicher Zeit berechnet und somit einen vielversprechenden Ausgangspunkt darstellt. Des Weiteren verfügen kompakte und konvexe Mengen im Allgemeinen über zahlreiche nützliche Eigenschaften [vgl. 17, S. 24, 146], auf die bei der Entwicklung der Lösungsmethode zurückgegriffen werden kann, siehe hierzu z. B. [5, Kapitel 2-4] oder [32, Kapitel 2.4]. Insbesondere lassen sich konkrete Aussagen über die Existenz und Qualität polyedrischer Approximationen treffen, wenn bei der zu approximierenden Menge Konvexität und Kompaktheit vorliegen. Beispielsweise existiert für eine beliebige kompakte und konvexe Menge C zu jedem $\epsilon > 0$ ein beschränktes Polyeder P , sodass

$$P \subseteq C$$

und zugleich C in der ϵ -Umgebung von P enthalten ist. Angesichts der zentralen Bedeutung beschränkter Polyeder für die Approximation konvexer Mengen erscheint der Einsatz polyedrischer Approximationen innerhalb der zu entwickelnden Lösungsmethode besonders geeignet [vgl. 27, S. 133f].

Die Arbeit ist wie folgt strukturiert: Zunächst werden in Abschnitt 2 grundlegende Notationen, Definitionen sowie zentrale mathematische Begriffe der Mengenoptimierung eingeführt. Darauf aufbauend werden Optimierer und ϵ -Optimierer als Lösungskonzepte mengenwertiger Optimierungsprobleme vorgestellt. Anschließend wird in Abschnitt 3 die Existenz von Optimierern für kompakt-konvexe Mengenoptimierungsprobleme nachgewiesen. In Abschnitt 4 wird ein Lösungsalgorithmus zur Berechnung von ϵ -Optimierern entwickelt, der auf bereits bestehenden Algorithmen basiert. Die Korrektheit sowie die Endlichkeit des Algorithmus werden in den Unterabschnitten 4.1 und 4.2 untersucht. Darüber hinaus werden in Abschnitt 5 zwei Beispiele implementiert und gelöst. Ausgewählte Implementierungsdetails werden ebenfalls präsentiert. Abschließend folgen eine Diskussion der erarbeiteten Ergebnisse sowie ein Ausblick auf mögliche weiterführende Forschungsrichtungen. Eine Zusammenfassung der gesamten Arbeit ist in Abschnitt 7 zu finden.

2 Mathematische Grundlagen

2.1 Notationen

Seien die Vektoren $x, y \in \mathbb{R}^n$ gegeben. Dann ist die *euklidische Norm* des Vektors x definiert durch

$$\|x\| = (x^T x)^{1/2} = (x_1^2 + \dots + x_n^2)^{1/2}$$

und die euklidische Distanz zwischen x und y definiert als $d(x, y) = \|x - y\|$ [vgl. 5, S. 633f]. Die Menge aller Vektoren der Länge kleiner gleich eins bezüglich der euklidischen Norm,

$$B = B_1(0) = \{x \in \mathbb{R}^n \mid \|x\| \leq 1\},$$

heißt euklidische *Einheitskugel* und $B_\epsilon(x) = x + \epsilon B$ beschreibt eine ϵ -*Kugel* mit Mittelpunkt x und Radius ϵ [vgl. 5, S. 633f, 36, S. 25]. Sei die Menge $C \subseteq \mathbb{R}^n$ gegeben. Ein $x \in C$ wird *innerer Punkt* der Menge C genannt, falls ein $\epsilon > 0$ existiert, sodass

$$\{y \in \mathbb{R}^n \mid \|y - x\| \leq \epsilon\} \subseteq C,$$

d. h. es gibt eine Kugel mit Mittelpunkt x , die gänzlich in C enthalten ist. Die Menge aller inneren Punkte von C heißt *Inneres* der Menge und wird im Folgenden mit $\text{int } C$ abgekürzt. C ist genau dann *abgeschlossen*, wenn der Grenzwert jeder konvergenten Folge in C liegt. D. h. falls $c_1, \dots, c_j, c_{j+1}, \dots$ gegen c konvergiert und alle $c_{j \in J} \in C$, dann ist $c \in C$. Der *Abschluss* der Menge C , $\text{cl } C$, beschreibt die Menge aller Grenzwerte konvergenter Folgen in C . Der *Rand* von C ist definiert durch

$$\text{bd } C = \text{cl } C \setminus \text{int } C$$

und ein Punkt $c \in \text{bd } C$ heißt *Randpunkt* der Menge C [vgl. 5, S. 638]. Die Menge C gilt als *beschränkt*, falls ein $\epsilon > 0$ existiert, sodass $C \subset B_\epsilon(0)$. Liegen sowohl Abgeschlossenheit als auch Beschränktheit vor, wird die Menge C *kompakt* genannt [vgl. 27, S. 5-8]. C heißt *konvex*, falls für $x, y \in C$ auch die Verbindungsstrecke

$$[x, y] = \{(1 - \lambda) \cdot x + \lambda \cdot y \mid 0 \leq \lambda \leq 1, \lambda \in \mathbb{R}\}$$

zu der Menge C gehört [vgl. 2, S. 11]. Die konvexe Hülle von C beschreibt die Menge aller endlichen Konvexitätskombinationen der Punkte in C [vgl. 5, S. 24]:

$$\text{conv } C = \{\lambda_1 x_1 + \dots + \lambda_k x_k \mid x_j \in C, \lambda_j \geq 0, j = 1, \dots, k, \lambda_1 + \dots + \lambda_k = 1\}.$$

Als kompakt-konvexer *Körper* wird eine kompakte und konvexe Menge C bezeichnet, wenn der Ursprung in ihrem Inneren liegt [vgl. 7, S. 39]. Die Menge C ist ein *Kegel*, falls $\lambda c \in C$ für alle $c \in C$ und $\lambda > 0$. Der Rezessionskegel von C ist definiert als die Menge aller Vektoren y , sodass $x + ty \in C$ für alle $x \in C$ und $t \geq 0$ [vgl. 35, S. 61]. Die konische Hülle von C beschreibt alle konischen Kombinationen der Punkte in C [vgl. 5, S. 25]:

$$\text{cone } C = \{\lambda_1 x_1 + \dots + \lambda_k x_k \mid x_j \in C, \lambda_j \geq 0, j = 1, \dots, k\}.$$

Die affine Hülle der Menge C ist diejenige Menge, die alle affinen Kombinationen von C enthält [vgl. 5, S. 23]:

$$\text{aff } C = \{\lambda_1 x_1 + \dots + \lambda_k x_k \mid x_j \in C, j = 1, \dots, k, \lambda_1 + \dots + \lambda_k = 1\}.$$

Eine *Hyperebene* ist eine Menge der Form $\{x \in \mathbb{R}^n \mid a^T x = b\}$ mit $a \in \mathbb{R}^n, a \neq 0$, $b \in \mathbb{R}$ und teilt den \mathbb{R}^n in zwei Halbräume. Ein abgeschlossener *Halbraum* ist die Menge $\{x \in \mathbb{R}^n \mid a^T x \leq b\}$, mit $a \in \mathbb{R}^n, a \neq 0$ und $b \in \mathbb{R}$ [vgl. 5, S. 27]. Ein *Polyeder* $P \subseteq \mathbb{R}^n$ ist definiert als die Lösungsmenge endlich vieler linearer Gleichungen und Ungleichungen [vgl. 5, S. 31]. Die *H-Darstellung* eines Polyeders

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\} \quad \text{mit} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

beschreibt die Formulierung von P als eine Schnittmenge endlich vieler abgeschlossener Halbräume, während die *V-Darstellung* den Polyeder als die Minkowski-Summe einer konvexen Hülle endlich vieler Punkte und einer konischen Hülle endlich vieler Richtungen darstellt:

$$P = \text{conv}(V) + \text{cone}(D) \quad \text{mit} \quad V \in \mathbb{R}^{n \times k}, D \in \mathbb{R}^{n \times l}.$$

Dabei ist die Minkowski-Summe von zwei Mengen $M_1, M_2 \subseteq \mathbb{R}^n$ definiert durch $M_1 + M_2 = \{x + y \mid x \in M_1, y \in M_2\}$ [vgl. 38, S. 28f]. Eine *Seite* eines Polyeders wird definiert als dessen Schnittmenge mit einer Hyperebene, bei der der Polyeder vollständig in einem der beiden entstehenden Halbräume enthalten ist. Seiten, die aus einem einzigen Punkt bestehen, heißen *Ecken* und Seiten, deren Dimension eins kleiner ist als die des Polyeders selbst, werden als *Facetten* bezeichnet. Die Dimension eines Polyeders ist gleich der Dimension seiner affinen Hülle [vgl. 38, S. 5]. Beschränkte Polyeder werden *Polytope* genannt [vgl. 5, S. 31].

2.2 Hausdorffabstand

In den nachfolgenden Abschnitten dieser Arbeit werden polyedrische Innenapproximationen kompakt-konvexer Mengen verwendet. Um die Güte der Approximationen bewerten zu können, wird die Distanz zwischen dem Polyeder und der zu approximierenden Menge untersucht. Eine weit verbreitete Methode dafür ist die Hausdorffdistanz, eine Metrik auf dem Raum der kompakten und nichtleeren Teilmengen des \mathbb{R}^n . Die Definition beruht auf dem Exzessbegriff [vgl. 7, S. xvii, 39f]. Für zwei Mengen $M_1, M_2 \subseteq \mathbb{R}^n$ wird der *Exzess* definiert als

$$e[M_1, M_2] = \sup_{x \in M_1} d(x, M_2)$$

und ist nicht symmetrisch bezüglich seiner Argumente. Ist die Menge M_1 beschränkt, dann ist $e[M_1, M_2]$ endlich, im Allgemeinen ist dies aber nicht der Fall. Wenn die Menge zusätzlich abgeschlossen ist, wird das Supremum von einem Punkt aus M_1 angenommen und kann mit dem Maximum ersetzt werden. Für die Mengen $M_1, M_2 \subseteq \mathbb{R}^n$ ist der *Hausdorffabstand zwischen M_1 und M_2* wie folgt definiert [vgl. 7, S. xvii, 39f]:

$$d_H(M_1, M_2) = \max\{e[M_1, M_2], e[M_2, M_1]\}.$$

Sind $M_1, M_2 \subseteq \mathbb{R}^n$ abgeschlossen und nichtleer, lässt sich der Hausdorffabstand alternativ unter Verwendung der Einheitskugel ausdrücken [vgl. 14, S. 762]:

$$d_H(M_1, M_2) = \inf\{\epsilon \geq 0 \mid M_1 \subseteq M_2 + \epsilon B, M_2 \subseteq M_1 + \epsilon B\}.$$

Es heißt, dass eine Sequenz $\{M_k\}_{k \in \mathbb{N}}$ aus Teilmengen des \mathbb{R}^n gegen eine Menge $M \subseteq \mathbb{R}^n$ *Hausdorff-konvergiert* oder *H-konvergiert*, falls

$$\lim_{k \rightarrow \infty} d_H(M_k, M) = 0.$$

Für alle konvexen Körper $C \subseteq \mathbb{R}^n$ existiert eine Sequenz aus Polytopen $\{P_k\}_{k \in \mathbb{N}}$, die gegen C H-konvergiert [vgl. 7, S. 40].

2.3 Mengenoptimierung

Eine *mengenwertige Abbildung* $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$ ist charakterisiert durch ihren *Graphen*

$$\text{gr } F = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^q \mid y \in F(x)\}$$

mit Definitionsbereich $\text{dom } F = \{x \in \mathbb{R}^n | F(x) \neq \emptyset\}$. Eine mengenwertige Funktion F besitzt eine bestimmte Eigenschaft, zum Beispiel Konvexität oder Kompaktheit, genau dann, wenn ihr Graph diese Eigenschaft ebenfalls aufweist [vgl. 3, S. 34ff]. Das Problem F bezüglich der Mengeninklusion \subseteq zu maximieren heißt *Mengenoptimierungsproblem* und hat im Allgemeinen folgende Form [vgl. 29, S. 331]:

$$\max \quad F(x) \quad \text{bzgl.} \quad x \in \mathbb{R}^n. \quad (\text{P})$$

In dieser Arbeit werden hauptsächlich polyedrisch-konvexe und kompakt-konvexe Mengenoptimierungsprobleme behandelt. Somit sind auch die jeweiligen Zielfunktionen polyedrisch-konvex bzw. kompakt-konvex, d. h. die Graphen stellen jeweils konvexe Polyeder bzw. kompakt-konvexe Mengen dar. Als Lösungskonzepte werden *Optimierer* [vgl. 28, 29] und ϵ -*Optimierer* verwendet.

Definition 2.1. Ein $\bar{x} \in \mathbb{R}^n$ heißt *Optimierer* [vgl. 28, S. 2], wenn

$$\nexists x \in \mathbb{R}^n : F(x) \supsetneq F(\bar{x}),$$

und ϵ -*Optimierer*, falls für ein $\epsilon \geq 0$:

$$\nexists x \in \mathbb{R}^n : F(x) \supsetneq F(\bar{x}) + \epsilon B.$$

Demnach ist ein Optimierer ein Punkt \bar{x} , dessen zugehöriger Funktionswert $F(\bar{x})$ bezüglich der Mengeninklusion \subseteq nicht mehr strikt vergrößert werden kann. Die zentrale Idee eines ϵ -Optimierers besteht hingegen darin, eine Optimierung bis auf eine vorgegebene Toleranz $\epsilon \geq 0$ zuzulassen. Statt zu verlangen, dass $F(\bar{x})$ von keinem anderen Funktionswert $F(x)$ eine strikte Teilmenge darstellt, wird lediglich gefordert, dass keine echte Verbesserung über eine ϵ -Umgebung hinaus möglich ist. Folglich ist $F(\bar{x})$ mit \bar{x} als ϵ -Optimierer bis auf die Genauigkeit ϵ maximal.

Des Weiteren folgt unmittelbar aus Definition 2.1, dass jeder Optimierer zugleich ein ϵ -Optimierer ist. Das Lösungskonzept der ϵ -Optimierer erlaubt somit eine kontrollierte Annäherung an einen Optimierer. Insbesondere im Hinblick auf algorithmische Berechnungen lassen sich auf diese Weise endliche Abbruchkriterien formulieren und polyedrische Approximationen konstruktiv einsetzen. In den nachfolgenden Abschnitten wird gezeigt, dass kompakt-konvexe Mengenoptimierungsprobleme Optimierer besitzen, und wie ϵ -Optimierer konkret konstruiert werden können.

3 Existenz von Optimierern

In diesem Abschnitt wird die Existenz von den in Definition 2.1 eingeführten Optimierern und ϵ -Optimierern thematisiert. Ziel ist es, unter den bereits getroffenen Annahmen der Kompaktheit und Konvexität der mengenwertigen Funktion $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$, zu zeigen, dass das Optimierungsproblem (P) mindestens einen ϵ -Optimierer besitzt. Der Existenzbeweis verwendet hierzu ein Resultat der Mengenlehre: das Lemma von Zorn 3.1 [vgl. 21, S. 356].

In dem Lemma von Zorn 3.1 werden Ketten und deren obere Schranken verwendet, um die Existenz eines maximalen Elements einer Menge nachzuweisen. Eine Kette $\mathcal{Z} = \{Z_j \mid j \in J\} \subseteq \mathcal{C}$ ist eine total geordnete Teilmenge einer halbgeordneten Menge \mathcal{C} , d. h. Ketten können wie folgt dargestellt werden:

$$Z_1 \subseteq Z_2 \subseteq \dots \subseteq Z_j \subseteq Z_{j+1} \subseteq \dots \quad \text{mit } Z_j \in \mathcal{Z}, \quad j \in J.$$

Die Kette $\mathcal{Z} \subseteq \mathcal{C}$ besitzt eine *obere Schranke* in \mathcal{C} , falls es ein $C \in \mathcal{C}$ gibt, sodass $Z \subseteq C$ für alle $Z \in \mathcal{Z}$ gilt. $\tilde{C} \in \mathcal{C}$ wird als *maximales Element* der Menge \mathcal{C} bezüglich der Halbordnung \subseteq bezeichnet, wenn für alle

$$C \in \mathcal{C} : \tilde{C} \subseteq C \Rightarrow \tilde{C} = C$$

gilt [vgl. 18, S. 6, 26, S. 170]. Oftmals wird zuerst die Vereinigung der einzelnen Kettenglieder als potentielle obere Schranke der Kette in Betracht gezogen [vgl. 21, S. 356, 26, S. 170].

Lemma 3.1. (*Lemma von Zorn [vgl. 18, S. 6]*) Sei \mathcal{C} eine halbgeordnete Menge. \mathcal{C} besitzt ein maximales Element, wenn jede Kette $\mathcal{Z} \subseteq \mathcal{C}$ eine obere Schranke in \mathcal{C} hat.

Proposition 3.2. Sei $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$ eine mengenwertige Abbildung, dessen Graph kompakt ist. Dann existiert ein Optimierer.

Beweis. Angenommen der Definitionsbereich der Funktion F sei leer. Dann ist für alle $\bar{x} \in \mathbb{R}^n$ der Funktionswert gleich der leeren Menge, d. h. $F(\bar{x}) = \emptyset$. Dementsprechend gilt, dass

$$\forall \bar{x} \in \mathbb{R}^n \nexists x \in \mathbb{R}^n : F(x) \supseteq F(\bar{x}),$$

weshalb jedes $\bar{x} \in \mathbb{R}^n$ einen Optimierer der Funktion F darstellt. Demnach existiert mindestens ein Optimierer, wodurch die Behauptung in diesem Fall erfüllt ist.

Angenommen der Definitionsbereich der Funktion F sei nichtleer. Zudem sei $\mathcal{B} = \{F(x) \mid x \in \mathbb{R}^n\}$ und $\mathcal{Z} = (Z_j)_{j \in J} \subseteq \mathcal{B}$ eine beliebige Kette mit einer beliebigen

Indexmenge J und der Mengeninklusion \subseteq als Halbordnung. Dann existiert eine Folge $(\tilde{x}_j)_j$, sodass

$$\begin{array}{ccccccccc} Z_1 & \subseteq & Z_2 & \subseteq & \dots & \subseteq & Z_j & \subseteq & \dots \text{ mit } j \in J, \quad Z_j \in \mathcal{Z} \\ \parallel & & \parallel & & & & \parallel & & \\ F(\tilde{x}_1) & \subseteq & F(\tilde{x}_2) & \subseteq & \dots & \subseteq & F(\tilde{x}_j) & \subseteq & \dots \text{ mit } j \in J, \quad F(\tilde{x}_j) \in \mathcal{B}. \end{array}$$

Die Folge der monoton wachsenden Teilmengen $(F(\tilde{x}_j))_j$ der Obermenge \mathcal{B} konvergiert und hat

$$F_\infty := \lim_{j \rightarrow \infty} F(\tilde{x}_j) = \bigcup_{j \in J} F(\tilde{x}_j)$$

als ihren Grenzwert [vgl. 9, S. 9]. Da der Definitionsbereich der Funktion F nichtleer ist, existiert mindestens ein $j \in J$ mit $F(\tilde{x}_j) \neq \emptyset$. Aufgrund der Monotonie der Kette sind dann alle nachfolgenden Funktionswerte ebenfalls nichtleer. Sei

$$\text{proj}_y(\text{gr } F) := \{y \in \mathbb{R}^q \mid \exists x \in \mathbb{R}^n : (x, y) \in \text{gr } F\}$$

die Projektion [vgl. 38, S. 32ff] des Graphen auf ihre y -Komponente. Wegen der Kompaktheit des Graphen, ist $\text{proj}_y(\text{gr } F)$ ebenfalls kompakt [vgl. 17, S. 26]. Es gilt, dass

$$F_\infty = \bigcup_{j \in J} F(\tilde{x}_j) \subseteq \bigcup_{x \in \text{dom } F} F(x) = \text{proj}_y(\text{gr } F).$$

Aus $F_\infty \subseteq \text{proj}_y(\text{gr } F)$ und der Kompaktheit der Projektion folgt die Beschränktheit der Menge F_∞ . Es gilt zu zeigen, dass ein $x_\infty \in \text{dom } F$ mit $F_\infty = F(x_\infty)$ existiert. Sei hierzu $(\tilde{x}_j, \tilde{y}_j)_j$ eine Folge mit $\tilde{y}_j \in F(\tilde{x}_j)$. Da der Graph von F kompakt ist, hat jede Folge $(\tilde{x}_j, \tilde{y}_j)_j$ eine konvergente Teilfolge: $(\tilde{x}_{j_k}, \tilde{y}_{j_k})_k \rightarrow (x_\infty, y_\infty)$ [vgl. 32, S. 60]. Aufgrund der Abgeschlossenheit des Graphen ist $(x_\infty, y_\infty) \in \text{gr } F$ [vgl. 5, S. 638]. Dadurch ist $x_\infty \in \text{dom } F$ und $F(x_\infty) \in \mathcal{B}$. Aufgrund des monotonen Wachstums der Mengen $F(\tilde{x}_j)$ gilt, dass

$$F(\tilde{x}_1) \subseteq F(\tilde{x}_2) \subseteq \dots \subseteq F(x_\infty)$$

Daraus folgt, dass $F_\infty \subseteq F(x_\infty)$. F_∞ ist als Vereinigung aller Mengen der Kette per Definition die kleinste obere Schranke der Kette. Deswegen kann es keine kleinere Menge in \mathcal{B} geben, die die Kette übertrifft, da F_∞ sonst nicht die Vereinigung aller Mengen ist. Deshalb muss $F(x_\infty) = F_\infty$ gelten, wodurch $F_\infty = F(x_\infty)$ folgt. Demnach existiert ein $F(x_\infty) \in \mathcal{B}$, das eine obere Schranke der Kette \mathcal{Z} bezüglich der Mengeninklusion \subseteq darstellt. Somit lässt sich das Lemma von Zorn 3.1 anwenden, woraus die Existenz eines Optimierers folgt. \square

Folgerung 3.3. Sei $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$ eine mengenwertige Abbildung, dessen Graph kompakt ist. Dann existiert ein ϵ -Optimierer.

Beweis. Die Aussage folgt unmittelbar aus der Definition des Optimierers und ϵ -Optimierers in Definition 2.1 sowie der Proposition 3.2. \square

Somit ist die Existenz der Optimierer und ϵ -Optimierer nicht nur für kompakt-konvexe Probleme, sondern für alle kompakten Mengenoptimierungsprobleme gesichert. Die Existenz eines ϵ -Optimierers lässt sich auch ohne die Verwendung des Lemmas von Zorn nachweisen. Dazu wird im nachfolgenden Abschnitt ein Algorithmus entwickelt, der in endlich vielen Schritten einen ϵ -Optimierer konstruiert. Da das Lemma von Zorn nicht konstruktiver Natur ist, wird dafür auf bereits existierende Lösungsansätze für mengenwertige, polyedrisch-konvexe Probleme zurückgegriffen, um darauf aufbauend einen Lösungsalgorithmus zu entwickeln.

4 Algorithmus zur Berechnung von ϵ -Optimierern

In diesem Abschnitt wird eines der Hauptresultate der Arbeit präsentiert: ein Algorithmus zur Berechnung eines ϵ -Optimierers von kompakt-konvexen Problemen. Dieser basiert auf dem Algorithmus 1 von Löhne [29], der für bestimmte polyedrisch-konvexe Probleme einen Optimierer berechnet, indem iterativ die Facetten eines Ausgangspolyeders entlang seiner Außennormalen maximal verschoben werden. Dabei ist das nachfolgende lineare Problem von zentraler Bedeutung. Es bestimmt einen Punkt, der die polyedrische Menge erweitert:

$$\max_{x,y} w^T y \quad \text{bzgl.} \quad \begin{cases} y & \in G(x) \\ G(\bar{x}) & \subseteq G(x), \end{cases} \quad (\text{LP}(G, w, \bar{x}))$$

wobei $G : \mathbb{R}^n \Rightarrow \mathbb{R}^q$ polyedrisch-konvex, $\bar{x} \in \text{dom } G$ und $w \in \mathbb{R}^q$ ist. Das Problem ist zulässig für alle $\bar{x} \in \text{dom } G$ und bestimmt anhand der Zielfunktion die maximal mögliche Ausdehnung von $G(\bar{x})$ in Richtung w . Dabei garantiert die erste Restriktion, dass y ein zulässiger Funktionswert von G ist, während die zweite Nebenbedingung sicherstellt, dass die Vergrößerung von $G(\bar{x})$ in Richtung w keine Verkleinerung in einer anderen Richtung hervorruft. Letzteres lässt sich mittels der V-Darstellung der polyedrischen Menge $G(\bar{x})$ überprüfen. Da alle Werte von G den selben Rezessionskegel besitzen, ist die zweite Restriktion genau dann erfüllt, wenn alle Punkte der V-Darstellung von $G(\bar{x})$ in $G(x)$ liegen [vgl. 29, S. 334f].

Darüber hinaus wird in Algorithmus 1 ein *minimales System der Außennormalen* polyedrisch-konvexer Mengen $P \subseteq \mathbb{R}^q$ verwendet. Das System besteht aus einer Familie von Vektoren $\{w_1, \dots, w_k\}$, die sich aus der Ungleichungsdarstellung

$$P = \{y \in \mathbb{R}^q \mid y^T w_1 \leq \gamma_1, \dots, y^T w_k \leq \gamma_k\} \quad \text{mit} \quad \|w_1\| = \dots = \|w_k\| = 1$$

und minimalem k ergeben. Wenn der Polyeder P ein nichtleeres Inneres hat, dann stellen die Vektoren w_j die eindeutig bestimmten normierten Außennormalen der Facetten von P dar [vgl. 29, S. 334].

In Algorithmus 1 werden die Facetten eines initialen Polyeders $G(\bar{x})$ möglichst weit nach außen verschoben bis keine weitere Vergrößerung möglich ist. Dazu wird für jede in Frage kommende Richtung w des minimalen Systems der Außennormalen N von $G(\bar{x})$ das Problem $(\text{LP}(G, w, \bar{x}))$ gelöst und getestet, ob die Menge in Richtung w echt vergrößert werden kann, d. h. $y^* \notin G(\bar{x})$. Ist dies der Fall, werden sowohl \bar{x} als auch N aktualisiert und der Vorgang wiederholt, bis alle Richtungen des derzeitigen Systems überprüft worden sind. Das schließlich zurückgegebene \bar{x} entspricht dem Optimierer der Funktion G [vgl. 29, S. 334f].

Algorithmus 1: Berechnung eines Optimierers \bar{x} mit $\bar{y} \in G(\bar{x})$ und G polyedrisch-konvex [vgl. 29, Algorithmus 1]

```

1 Function ComputeOptimizer( $G, \bar{y}$ ):
2   wähle ein  $\bar{x} \in \mathbb{R}^n$  mit  $\bar{y} \in G(\bar{x})$ 
3    $N \leftarrow \{\text{minimales System der Außennormalen von } G(\bar{x})\}$ 
4    $W \leftarrow \emptyset$ 
5   repeat
6     wähle ein  $w \in N \setminus W$ 
7      $W \leftarrow W \cup \{w\}$ 
8     if ( $LP(G, w, \bar{x})$ ) hat eine Optimallösung then
9       | weise sie  $(x^*, y^*)$  zu
10      else
11        | return  $\emptyset$ 
12      end
13      if  $y^* \notin G(\bar{x})$  then
14        |  $\bar{x} \leftarrow x^*$ 
15        |  $N \leftarrow \{\text{minimales System der Außennormalen von } G(\bar{x})\}$ 
16      end
17    until  $N \subseteq W$ 
18 return  $\{\bar{x}\}$ 

```

Algorithmus 2 beruht auf demselben Kernkonzept wie Algorithmus 1. Allerdings müssen bei der Übertragung des Ansatzes auf kompakt-konvexe Probleme mehrere Anpassungen getroffen werden. Angefangen bei dem linearen Problem $(LP(G, w, \bar{x}))$ lässt sich die zweite Nebenbedingung für kompakt-konvexe, mengenwertige Funktionen im Allgemeinen nicht analog nachprüfen, da im Gegensatz zu Polyedern bei der V-Darstellung nicht-polyedrischer, konvexer Mengen unendlich viele Punkte vorliegen können [vgl. 5, S. 31f, 38, S. 29f]. Deshalb wird stattdessen in Algorithmus 2 das nachfolgende konvexe Problem verwendet:

$$\max_{x,y} w^T y \quad \text{bzgl.} \quad \begin{cases} y & \in F(x) \\ I_{F(\bar{x})} & \subseteq F(x). \end{cases} \quad (\text{CP}(F, w, I_{F(\bar{x}})))$$

$I_{F(\bar{x})}$ bezeichnet eine polyedrische Innenapproximation von $F(\bar{x})$. Die Zielfunktion und die erste Restriktion von $(\text{CP}(F, w, I_{F(\bar{x}})))$ bleiben im Vergleich zu $(LP(G, w, \bar{x}))$ strukturell identisch, wobei statt der polyedrisch-konvexen, mengenwertigen Funktion G die kompakt-konvexe Funktion $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$ genutzt wird. Dahingegen verwendet die zweite, abgeänderte Nebenbedingung eine polyedrische Innenapproximation $I_{F(\bar{x})}$ statt der konvexen Menge $F(\bar{x})$ selbst. Dadurch wird gewährleistet, dass Verbesserungen in einer Richtung lediglich zu kontrollierten Verschlechterungen in anderen Richtungen führen können und zugleich die zweite Nebenbedingung analog zu denjenigen des linearen Problems $(LP(G, w, \bar{x}))$ überprüfbar bleibt.

Algorithmus 2: Berechnung eines ϵ -Optimierers \bar{x} mit $\bar{y} \in F(\bar{x})$ in Anlehnung an Algorithmus 1 aus [29]

```

1 Function ComputeEpsilonOptimizer( $F, \bar{y}, \epsilon$ ):
2   wähle ein  $\bar{x} \in \mathbb{R}^n$  mit  $\bar{y} \in F(\bar{x})$ 
3    $I \leftarrow \{\text{Innenapproximation von } F(\bar{x})\}$ 
4    $N \leftarrow \{\text{minimales System der Außennormalen von } I\}$ 
5    $W \leftarrow \emptyset$ 
6   repeat
7     wähle ein  $w^* \in \arg \max_{w \in N \setminus W} (CP(F, w, I))$  mit  $(x^*(w^*), y^*(w^*))$ 
8      $W \leftarrow W \cup \{w^*\}$ 
9     if  $y^*(w^*) \notin I + \epsilon B$  then
10    |    $\bar{x} \leftarrow x^*(w^*)$ 
11    |    $I \leftarrow \{\text{Innenapproximation von } F(\bar{x})\}$ 
12    |    $N \leftarrow \{\text{minimales System der Außennormalen von } I\}$ 
13    end
14   until  $N \subseteq W$ 
15 return  $\{\bar{x}\}$ 

```

Ähnlich wie in Algorithmus 1 beginnt die Funktion *ComputeEpsilonOptimizer* des Algorithmus 2 mit der Initialisierung einer konvexen Menge $F(\bar{x})$, der zugehörigen polyedrischen Innenapproximation I und deren minimalem System der Außennormalen N , sowie der Menge W , die die bereits überprüften Richtungen $w \in N$ enthält. Die anschließende Schleife wird solange durchlaufen, bis alle Richtungen w des aktuellen Systems N überprüft worden sind, d. h. $N \subseteq W$. Im Vergleich zu Algorithmus 1 enthält Algorithmus 2 keine weiteren Abbruchbedingungen.

In der Schleife wird zuerst das konvexe Problem $(CP(F, w, I_{F(\bar{x}})))$ für alle noch nicht überprüften Richtungen $w \in N \setminus W$ gelöst, eine Richtung w^* mit dem höchsten Optimalwert gewählt und die dazugehörigen Optimallösungen in $(x^*(w^*), y^*(w^*))$ abgespeichert. Dabei ist anzumerken, dass für jedes konvexe Problem $(CP(F, w, I_{F(\bar{x}})))$ stets eine Optimallösung existiert (siehe Lemma 4.2). Danach wird w^* der Menge der überprüften Richtungen W hinzugefügt. Anschließend wird getestet, ob die Optimallösung $y^*(w^*)$ in $I + \epsilon B$ liegt, d. h. es wird überprüft, ob die Innenapproximation um mehr als ϵ in Richtung w^* vergrößert werden kann. Ist dies der Fall, werden \bar{x}, I und N nacheinander aktualisiert.

Sind alle Richtungen getestet worden, wird das aktuelle \bar{x} zurückgegeben. In den nachfolgenden Unterabschnitten wird aufgezeigt, dass es sich bei dem zurückgegebenen \bar{x} korrekterweise um einen ϵ -Optimierer von F handelt. Darüber hinaus wird die Endlichkeit des Algorithmus 2 thematisiert.

4.1 Korrektheit

Für den Beweis der Korrektheit des zuvor präsentierten Algorithmus 2 wird benötigt, dass das konvexe Optimierungsproblem $(CP(F, w, I_{F(\bar{x}})))$, unter den getroffenen Annahmen der Konvexität und Kompaktheit der mengenwertigen Funktion F , stets eine Optimallösung besitzt. Dazu wird der Extremwertsatz 4.1 verwendet.

Satz 4.1. (Extremwertsatz [vgl. 34, Theorem 9.3.2, S. 274]) Sei $D \subseteq \mathbb{R}^q$ kompakt und $f : D \rightarrow \mathbb{R}$ stetig, dann nimmt die Funktion f sowohl ihr Maximum als auch ihr Minimum auf D an.

Lemma 4.2. Sei $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$ eine kompakt-konvexe mengenwertige Abbildung mit nichtleerem Definitionsbereich, $w \in \mathbb{R}^q$, $\bar{x} \in \text{dom } F$ und $I_{F(\bar{x})}$ eine Innenapproximation von $F(\bar{x})$ mit $I_{F(\bar{x})} \subseteq F(\bar{x})$. Dann hat das $(CP(F, w, I_{F(\bar{x}})))$ eine Optimallösung.

Beweis. Der zulässige Bereich $D := \{(x, y) \mid x \in \text{dom } F, y \in F(x), I_{F(\bar{x})} \subseteq F(x)\}$ des Problems $(CP(F, w, I_{F(\bar{x}})))$ ist kompakt, da $F(x)$ kompakt ist, die Inkluisionsbedingung $I_{F(\bar{x})} \subseteq F(x)$ eine abgeschlossene Teilmenge definiert und der Schnitt kompakter Mengen mit abgeschlossenen Mengen wiederum kompakt ist [vgl. 23, S. 3, 10, S. 30-34]. Die stetige Zielfunktion $f(y) = w^T y$ nimmt daher nach Satz 4.1 ihr Maximum auf D an. Damit besitzt $(CP(F, w, I_{F(\bar{x}})))$ stets eine Optimallösung. \square

Proposition 4.3. Für eine beliebige kompakt-konvexe mengenwertige Abbildung $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^q$ mit nichtleerem Definitionsbereich, einen Punkt $\bar{y} \in F(\bar{x})$ und einen vorgegebenen Approximationsfehler $\epsilon \geq 0$ gibt die Funktion `ComputeEpsilonOptimizer` in Algorithmus 2 einen ϵ -Optimierer \bar{x} des Problems (P) mit der Eigenschaft $\bar{y} \in F(\bar{x})$ zurück.

Beweis. Angenommen, das zurückgegebene \bar{x} sei kein ϵ -Optimierer. Daraus folgt, dass

$$\exists \tilde{x} \in \mathbb{R}^n : F(\tilde{x}) \supseteq F(\bar{x}) + \epsilon B \quad \text{und} \quad \exists \tilde{y} \in F(\tilde{x}) : \tilde{y} \notin F(\bar{x}) + \epsilon B. \quad (1)$$

Nach einer regulären Terminierung der Iterationsschleife, stellt die Menge $N \subseteq W$ ein System der normierten Außennormalen von $I_{F(\bar{x})}$ dar. Demnach existiert ein $\tilde{w} \in N$, sodass

$$\tilde{w}^T \tilde{y} > \sup\{\tilde{w}^T z \mid z \in F(\bar{x}) + \epsilon B\} =: \tilde{\gamma}. \quad (2)$$

Da $\tilde{w} \in W$ ist, muss ein entsprechendes $(\text{CP}(F, \tilde{w}, I_{F(\bar{x}(\tilde{w}))}))$ in einer vorherigen Iteration gelöst worden sein. Dessen Optimallösung, die aufgrund von Lemma 4.2 existiert, wird im Rahmen dieses Beweises mit $(x^*(\tilde{w}), y^*(\tilde{w}))$ bezeichnet. Aus der zweiten Nebenbedingung des konvexen Optimierungsproblems ergibt sich die folgende Inklusion:

$$I_{F(\bar{x}(\tilde{w}))} \subseteq F(\bar{x}). \quad (3)$$

Zunächst wird gezeigt, dass

$$y^*(\tilde{w}) \in F(\bar{x}) + \epsilon B. \quad (4)$$

Falls $y^*(\tilde{w}) \in I_{F(\bar{x}(\tilde{w}))} + \epsilon B$ gilt, folgt die Aussage (4) unmittelbar aus (3). Ist allerdings $y^*(\tilde{w}) \notin I_{F(\bar{x}(\tilde{w}))} + \epsilon B$, so wird im Algorithmus 2 der Aktualisierungsschritt in Zeile 9 ausgeführt, woraufhin

$$y^*(\tilde{w}) \in F(x^*(\tilde{w})) \subseteq F(\bar{x}) + \epsilon B \quad (5)$$

gilt. Die Inklusionsbeziehung in (5) folgt, wie bereits (3), aus den Restriktionen des konvexen Problems:

$$I_{F(x^*(\tilde{w}))} \subseteq F(\bar{x}) \quad \Rightarrow \quad F(x^*(\tilde{w})) \subseteq I_{F(x^*(\tilde{w}))} + \epsilon B \subseteq F(\bar{x}) + \epsilon B.$$

Damit ist (4) erfüllt und impliziert, dass der Optimalwert $\tilde{w}^T y^*(\tilde{w})$ des Problems $(\text{CP}(F, \tilde{w}, I_{F(\bar{x}(\tilde{w}))}))$ nicht größer als $\tilde{\gamma}$ aus (2) ist. Außerdem folgt gemäß (1) und (3), dass (\tilde{x}, \tilde{y}) für das $(\text{CP}(F, \tilde{w}, I_{F(\bar{x}(\tilde{w}))}))$ zulässig ist. Somit besteht ein Widerspruch zu der anfänglich getroffenen Annahme, da der Wert $\tilde{w}^T \tilde{y}$ größer ist als der Optimalwert [vgl. 29, S. 335f]. \square

4.2 Endlichkeit

Im nachfolgenden Beweis der Endlichkeit des Algorithmus 2 wird auf ausgewählte Konvergenzresultate adaptiver Approximationsschemata nach Kamenev [19, 20] zurückgegriffen. Dazu werden in diesem Abschnitt zuerst die Initialisierung und die Aktualisierung der in Zeile 3 und Zeile 11 verwendeten Innenapproximation I spezifiziert, sodass Algorithmus 2 die Charakteristika eines sogenannten Augmenting-Schemas erfüllt. Ein *Augmenting-Schema* beschreibt eine Klasse polyedrischer Innenapproximationsalgorithmen für konvexe Körper, in der eine bekannte polyedrische Approximation jede Iteration verfeinert bzw. verbessert wird. Sei hierfür $C \subseteq \mathbb{R}^n$

ein konvexer Körper und I_k mit $I_k \subseteq C$ eine polyedrische Approximation. Dann ist eine Iteration eines Augmenting-Schemas definiert durch die nachfolgenden beiden Schritte [vgl. 7, S. 41f]:

1. Wähle einen Punkt p auf dem Rand von C .
2. Konstruiere $I_{k+1} = \text{conv}(I_k \cup \{p\})$.

Kamenev leitet in [19, 20] Konvergenzresultate für eine spezifische Unterklasse an Augmenting-Schemata her: die Hausdorff-Schemata. Ein Algorithmus erfüllt die Bedingungen eines *Hausdorff-Schemas* [vgl. 7, S. 42], falls ein $\gamma > 0$ existiert, sodass für jede Iteration k

$$d_H(I_k, I_{k+1}) \geq \gamma d_H(I_k, C).$$

Theorem 4.4. ([vgl. 7, Theorem 3.5, 19, Theorem 2]) Sei $C \subseteq \mathbb{R}^n$ ein konvexer Körper und $\{I_k\}_{k \in \mathbb{N}}$ eine Folge von Polyedern, generiert durch ein Hausdorff-Schema. Dann existiert für jedes $\mathcal{K} > 0$ ein k_0 , sodass für alle $k \geq k_0$: $d_H(I_k, C) \leq (1 + \mathcal{K}) \frac{c(C)}{k^{1/(n-1)}}$, wobei $c(C)$ eine Konstante darstellt, die von C abhängt.

Es wird im Folgenden angenommen, dass die Mengen $F(x)$ mit $x \in \text{dom } F$ kompakt-konvexe Körper sind. Dies stellt keine starke Einschränkung dar: ist $F(x)$ volldimensional, kann ein innerer Punkt $\bar{y} \in \text{int } F(x)$ berechnet werden und statt $F(x)$ die Menge $F(x) - \bar{y}$ verwendet werden, da diese den Ursprung in ihrem Inneren enthält. Ist $F(x)$ nicht volldimensional, kann der Umgebungsraum auf die affine Hülle von $F(x)$ beschränkt werden [vgl. 7, S. 49]. Ein Algorithmus zur Bestimmung innerer Punkte bei kompakt-konvexen Mengen befindet sich z. B. in [30, Kapitel 5.5]. Des Weiteren müssen für die benötigten Konvergenz- und Endlichkeitsaussagen aus [19, 20] sowohl die Initialisierung in Zeile 3 als auch die Aktualisierung der Approximation in Zeile 11 des Algorithmus 2 konkretisiert werden [vgl. 7, S. 53ff]. Dafür werden jeweils die Funktionen aus den Algorithmen 3 und 4 genutzt. Unter deren Verwendung erfüllt Algorithmus 2 die Kriterien eines Augmenting-Schemas. Kern des Algorithmus 3 ist das Problem $(\text{IP}(F, \bar{x}, p, d))$ mit $p, d \in \mathbb{R}^q$.

$$\max t \quad \text{bzgl.} \quad \begin{cases} y \in F(\bar{x}) \\ y = p + td \end{cases} \quad (\text{IP}(F, \bar{x}, p, d))$$

Im Allgemeinen ist $p \in F(\bar{x})$ ein Punkt und d eine Richtung. Das Problem $(\text{IP}(F, \bar{x}, p, d))$ beschreibt das Folgende: Ausgehend von $p \in F(\bar{x})$ wird die maximale Distanz in Richtung d bestimmt, die zurückgelegt werden kann, ohne die Menge $F(\bar{x})$ zu verlassen. Handelt es sich bei $F(\bar{x})$ um eine kompakte Menge, wird die maximale Distanz in dem Punkt erreicht, wo die Halbgerade $\{p + td \mid t > 0\}$ den Rand von $F(\bar{x})$ schneidet [vgl. 7, S. 44].

Algorithmus 3: Berechnung einer Innenapproximation I eines konvexen Körpers $F(\bar{x})$ [vgl. 7, Algorithmus 3.3]

```

1 Function ComputeInitialApproximation( $F(\bar{x})$ ):
2   |  $I \leftarrow \emptyset$ 
3   | for  $d \in \{-e, e_1, \dots, e_n\}$  do
4   |   | bestimme die Lösung  $(t_d, y_d)$  von  $(\text{IP}(F, \bar{x}, 0, d))$ 
5   |   |  $I \leftarrow \text{conv}(I \cup \{y_d\})$ 
6   | end
7 return  $I$ 
```

Im ersten Schritt der Funktion *ComputeInitialApproximation* in Algorithmus 3 wird die derzeitige Approximation mit der leeren Menge initialisiert. Dann wird eine initiale Innenapproximation I berechnet, indem $(\text{IP}(F, \bar{x}, 0, d))$ für $d = -e$ und alle Richtungen $d = e_j, j = 1, \dots, n$ gelöst wird. Der Vektor e_j beschreibt den j -ten Einheitsvektor, d. h. einen Nullvektor mit einer eins an j -ter Stelle, und e ein Vektor dessen Einträge alle gleich eins sind. Da $0 \in \text{int } F(\bar{x})$, erzeugt jede Lösung (t_d, y_d) den Randpunkt $y_d = t_d d$ von $F(\bar{x})$, wobei $t_d > 0$. Die initiale Approximation I wird konstruiert als die konvexe Hülle der $n + 1$ bestimmten Randpunkte [vgl. 7, S. 53]. Es kann nachgewiesen werden, dass diese Randpunkte $y_j, j = 0, \dots, n$ affin unabhängig sind, d. h. $y_1 - y_0, \dots, y_n - y_0$ sind linear unabhängig [vgl. 5, S.32f]. Dann hat die polyedrische Innenapproximation die Dimension n und ist somit ein volldimensionales, nichtleeres Polyeder innerhalb der Menge $F(\bar{x})$ [vgl. 7, S. 53].

Algorithmus 4: Aktualisierung einer Innenapproximation I eines konvexen Körpers $F(\bar{x})$ mit $y^* \in \text{bd } F(\bar{x})$ [vgl. 7, Algorithmus 3.3]

```

1 Function ComputeUpdatedApproximation( $I, y^*$ ):
2   |  $I \leftarrow \text{conv}(I \cup \{y^*\})$ 
3 return  $I$ 
```

Die Innenapproximation wird in Algorithmus 4 aktualisiert, indem das berechnete y^* aus $(\text{CP}(F, w, I_{F(\bar{x})}))$ der V-Darstellung der derzeitigen Approximation I hinzugefügt wird, falls die zugehörige Facette um mehr als ϵ innerhalb der Menge $F(\bar{x})$ verschoben werden kann. Dabei handelt es sich bei y^* um einen Randpunkt von $F(\bar{x})$, wodurch die neue Approximation $I_{k+1} = \text{conv}(I_k \cup \{y^*\})$ weiterhin eine Teilmenge von $F(\bar{x})$ darstellt [vgl. 7, S. 53ff].

Außerdem verwendet der Beweis der Endlichkeit des Algorithmus 2 das Konzept einer ϵ -Hausdorff-Approximation. Für eine konvexe Menge $C \subseteq \mathbb{R}^n$ und $\epsilon \geq 0$ heißt ein Polyeder $P \subseteq \mathbb{R}^n$ ϵ -Hausdorff-Approximation oder auch ϵ -H-Approximation von C , falls $d_H(P, C) \leq \epsilon$ [vgl. 7, S. 49].

Proposition 4.5. *Algorithmus 2 terminiert, unter Verwendung von Algorithmus 3 in Zeile 3, Algorithmus 4 in Zeile 11 und den Voraussetzungen aus Proposition 4.3, mit einer $\gamma\epsilon$ -H-Approximation I des kompakt-konvexen Körpers $F(\bar{x}) \subseteq \mathbb{R}^n$ für ein $\gamma \geq 0$. Darüber hinaus verkleinert sich $d_H(I_k, F(\bar{x}))$ in der Größenordnung $O(\frac{1}{k^{1/(n-1)}})$.*

Beweis. Für alle $d \in \{-e, e_1, \dots, e_n\}$ existieren die Lösungen (t_d, y_d) des Problems $(IP(F, \bar{x}, p, d))$, da $F(\bar{x})$ kompakt ist und $0 \in \text{int } F(\bar{x})$. Außerdem ist $y_d = t_d d$ ein Randpunkt und $t_d > 0$. Deshalb handelt es sich bei der Menge I , die in Zeile 3 mittels des Algorithmus 3 berechnet wird, um eine volldimensionale Innenapproximation von $F(\bar{x})$. Sei I_k die Approximation nach k Iterationen der Schleife in Zeile 6 - 14 des Algorithmus 2. Eine Lösung zu $(CP(F, w, I_{F(\bar{x}}))$ existiert nach Lemma 4.2 und das verwendete y^* in Zeile 11 ist ein Randpunkt von $F(x^*)$. Daher gilt für die aktualisierte Innenapproximation $I_{k+1} \subseteq F(x^*)$. Angenommen der Algorithmus terminiert nach K Iterationen mit I_K und α sei die größte Distanz, die eine Facette von I_K innerhalb der Menge $F(\bar{x})$ verschoben werden kann. Dann ist

$$\alpha = \max\{\sigma_{F(\bar{x})}(w) - \sigma_{I_K}(w) \mid w \in N\} \leq \epsilon,$$

da $N \subseteq W$ und sowohl I_K als auch die zu approximierende Menge $F(\bar{x})$ kompakt und konvex sind [vgl. 7, S. 55]. Hierbei bezeichnet $\sigma_C(c_1) = \sup\{c_1^T c_2 \mid c_2 \in C\}$ die Stützfunktion einer Menge C [vgl. 5, S. 81]. Des Weiteren ist nach [20, Theorem 1] der Hausdorffabstand zwischen I_K und $F(\bar{x})$ nach oben begrenzt durch

$$d_H(I_K, F(\bar{x})) \leq \frac{R}{r}\alpha,$$

wobei $B_r(0) \subseteq I_0 \subseteq F(\bar{x}) \subseteq B_R(0)$ und I_0 die initiale Innenapproximation ist, die von Algorithmus 3 berechnet wird. Daraus folgt, dass

$$\exists 0 < r < R : d_H(I_K, F(\bar{x})) \leq \frac{R}{r}\alpha \leq \frac{R}{r}\epsilon,$$

weshalb I_K eine $\gamma\epsilon$ -H-Innenapproximation mit $\gamma = \frac{R}{r}$ von $F(\bar{x})$ darstellt. Die Konvergenzrate ergibt sich direkt aus [20, Theorem 2] und Theorem 4.4. Außerdem impliziert dies, dass Algorithmus 2 nach endlich vielen Schritten terminiert [vgl. 7, S. 55]. \square

Anmerkung 4.6. *Es ist zu beachten, dass nach der Terminierung des Algorithmus 2 eine $\gamma\epsilon$ -H-Approximation vorliegt und im Allgemeinen keine ϵ -H-Approximation. Das bedeutet, dass der Hausdorffabstand zwischen der Approximation und $F(\bar{x})$ größer gleich ϵ sein kann, da $\gamma \geq 0$. Für eine exakte ϵ -H-Approximation müsste $\frac{\epsilon}{\gamma}$ als*

entsprechender Fehler in Algorithmus 2 gewählt werden. Zudem ließe sich für ein bekanntes γ die genaue Anzahl der notwendigen Iterationen zur Terminierung des Algorithmus 2 ermitteln. Allerdings ist ebendieses γ in der Praxis schwer zu berechnen, da die Variable von der konkreten Menge $F(\bar{x})$ und somit von der verwendeten Funktion F abhängig ist [vgl. 7, S. 42].

5 Beispiele

Im Folgenden werden zwei Beispiele betrachtet, anhand derer die Funktionalität und Funktionsweise des Algorithmus 2 aufgezeigt werden soll. Anschließend wird auf ausgewählte Implementierungsdetails eingegangen.

5.1 Wachsender Kreis

Beispiel 5.1. Sei die Funktion $F(x) : \mathbb{R} \rightrightarrows \mathbb{R}^2$ der Form

$$F(x) = \{y \in \mathbb{R}^2 \mid \|y\| \leq 1 + 0.3x\}, \quad x \in [0, 2] \cap \mathbb{R}$$

gegeben. Eine Implementierung des Algorithmus 2 berechnet für die Funktion F den Startpunkt $\bar{x} = 0.9960$, wenn vorgegeben wird, dass $\bar{y} = (0, 0)^T \in F(\bar{x})$ gelten soll. In Abbildung 1 befindet sich in der linken Grafik eine Illustration der Ausgangslage: die initial erzeugte Innenapproximation I (blau) von $F(\bar{x})$ (Rand in schwarz) und der ersten Lösung y^* von $(CP(F, w, I_{F(\bar{x})}))$ (rot). In analoger Darstellungsweise sind in den restlichen Grafiken der Abbildung 1 die weiteren Iterationen der Schleife des Algorithmus 2 abgebildet. Bereits nach der ersten Iteration stellt das aktuelle $\bar{x} = 2$ den ϵ -Optimierer dar. In Abbildung 2 sind jeweils die Endapproximationen für $\epsilon = 0.1$ und $\epsilon = 0.01$ abgebildet.

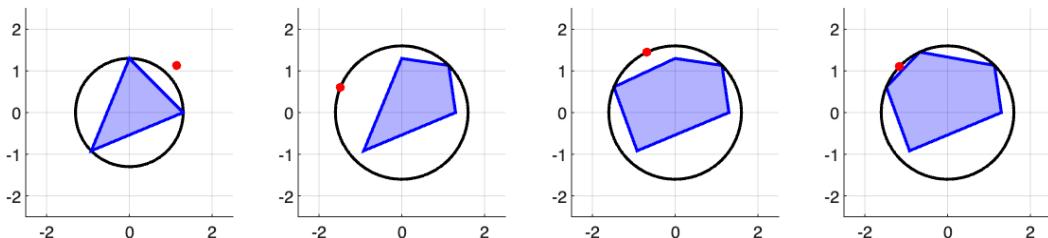


Abbildung 1: Visualisierung der ersten Iterationen der Schleife in Algorithmus 2 für Beispiel 5.1. In blau ist die aktuelle Innenapproximation gekennzeichnet, der Rand von $F(\bar{x})$ des derzeitigen \bar{x} in schwarz und der rote Punkt markiert in jeder Grafik die Lösung des zugehörigen $(CP(F, w, I_{F(\bar{x})}))$.

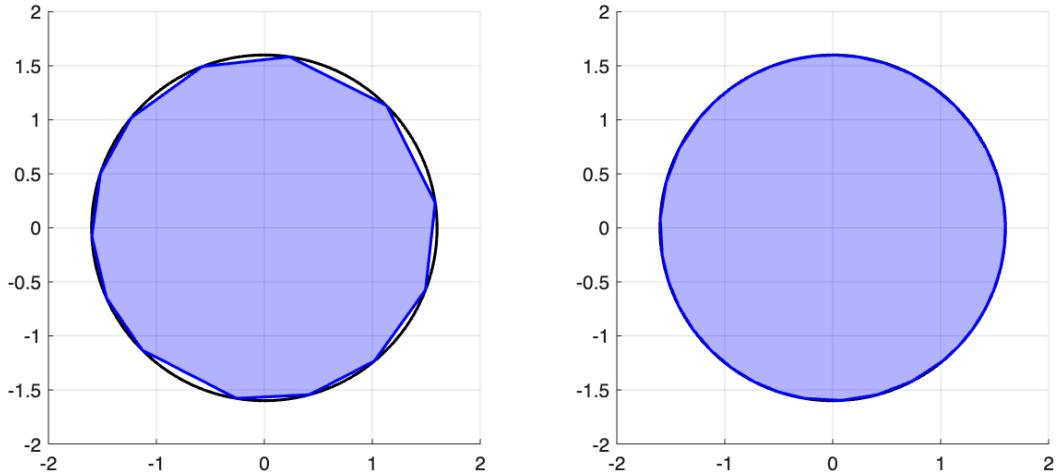


Abbildung 2: Berechnete Innenapproximationen für die Funktion aus Beispiel 5.1 mit $\epsilon = 0.1$ und $\epsilon = 0.01$, jeweils mit 13 bzw. 40 Ecken.

5.2 Portfoliooptimierung

Das zweite Beispiel ist inspiriert durch ein multikriterielles Problem aus [33, 37, Beispiel 5.1] und thematisiert die Portfoliooptimierung. Das Ziel besteht darin, das vorhandene Kapital so aufzuteilen, dass die erwartete Rendite des Portfolios maximiert wird, während das Risiko in Form der Varianz minimiert wird.

Angenommen es seien $n > 1$ Assets gegeben. Dann müssen die Investierenden entscheiden, wie sie ihr Kapital aufteilen, d. h. es muss eine passende Investitionsstrategie $x \in \mathbb{R}^n$ mit $x \geq 0$ und $\sum_{j=1}^n x_j = 1$ gewählt werden. x stellt somit einen Vektor aus Prozentsätzen dar: x_j gibt an, wie viel des gesamten Kapitals in Asset j investiert wird. Die erwartete Rendite der n -Assets sei durch $r \in \mathbb{R}^n$ angegeben, d. h. r_j entspricht der erwarteten Rendite des j -ten Assets, und $Q \in \mathbb{R}^{n \times n}$ sei die zugehörige Kovarianzmatrix. Dann ergibt sich das nachfolgende konvexe multikriterielle Problem mit der Zielfunktion $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^2$ [vgl. 37, S. 19].

$$\max_{x \in \mathbb{R}^n} \underbrace{(r^T x, -x^T Q x)^T}_{f(x)} \quad \text{bzgl.} \quad \begin{cases} \sum_{j=1}^n x_j = 1 \\ x \geq 0 \end{cases} \quad (\text{PO}(Q, r))$$

Q ist hierbei positiv semidefinit [vgl. 37, S. 19]. Eine Matrix $Q \in \mathbb{R}^{n \times n}$ heißt positiv semidefinit, falls für alle $x \neq 0 : x^T Q x > 0$ und Q symmetrisch ist, d. h. $Q^T = Q$ [vgl. 5, S. 646f]. Multikriterielle Probleme können beispielsweise mittels der Pareto-Optimalität gelöst werden. Eine zulässige Strategie heißt pareto-optimal, wenn es keine andere zulässige Strategie gibt, die ein komponentenweise strikt besseres Ergebnis erzeugt [vgl. 5, S. 57, 177].

Das Problem $(PO(Q, r))$ wird im Folgenden mengenwertig erweitert, um ein zweistufiges, konkaves Mengenoptimierungsproblem zu konstruieren. Bei zweistufigen Problemen wird in der ersten Stufe eine Entscheidung auf Basis von bekannten und unbekannten Variablen getroffen. Erst in der zweiten Stufe realisieren sich die unbekannten Variablen. Der Optimalwert hängt sowohl von der Zwischenentscheidung als auch von den nachträglich realisierten Variablen ab [vgl. 12, S. 1f]. Hierbei kann der Entscheidungsträger eine Präferenz für Flexibilität haben [vgl. 25, S. 565]. Es könnte ein Interesse darin bestehen, die Entscheidungen der zweiten Stufe flexibel zu halten, um auf die Realisierung der unbekannten Variablen reagieren zu können. D. h. die verfügbare Entscheidungsmenge soll größtmöglich sein. Das Problem $(MPO(Q, r, \delta, V))$ kann verwendet werden, um diese Präferenz für Flexibilität abzubilden, und bietet damit eine Alternative zu klassischen Lösungsansätzen multikriterieller Probleme, wie z. B. der Pareto-Optimalität [vgl. 12, S. 1f].

$$\max_{x \in \mathbb{R}^n} \underbrace{\text{conv} (f(x) + \delta(x)V) - \mathbb{R}_+^2}_{F(x)} \quad \text{bzgl.} \quad \begin{cases} \sum_{j=1}^n x_j = 1 \\ x \geq 0 \end{cases} \quad (MPO(Q, r, \delta, V))$$

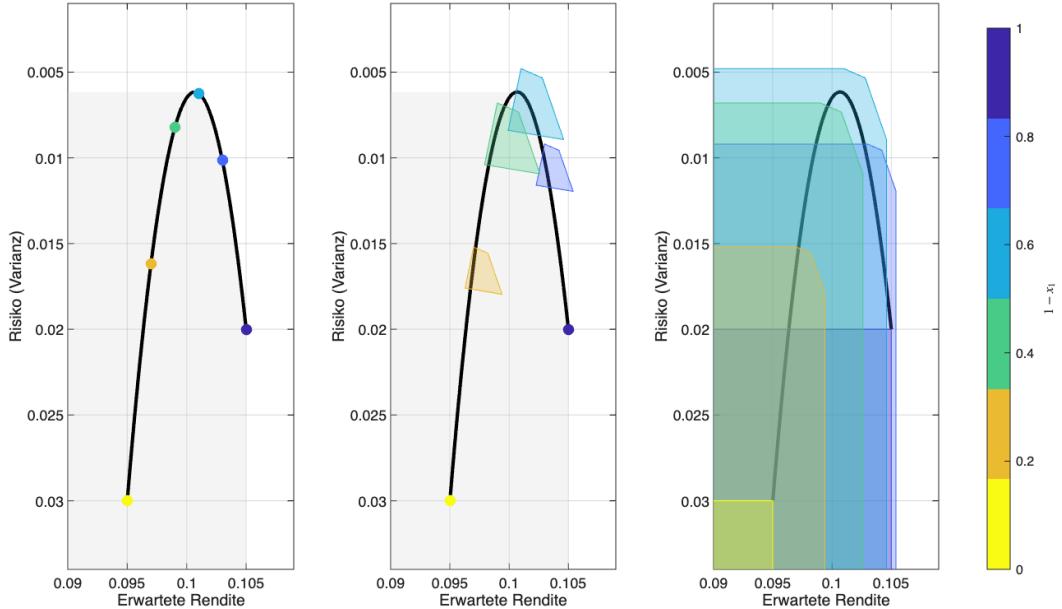


Abbildung 3: Illustrierung der Konstruktion von $F(x)$ aus $(MPO(Q, r, \delta, V))$ bei zwei Assets für fünf Investitionsstrategien $x = (x_1, 1 - x_1)^T$ mit $x_1 = \{0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1\}$. In der linken Grafik ist $f(x)$ abgebildet, in der Mitte $\text{conv}(f(x) + \delta(x)V)$ und rechts $F(x)$, jeweils mit den Werten aus Beispiel 5.2. Die schwarze Kurve beschreibt $f(x)$ für jede zulässige Strategie x und die graue Fläche repräsentiert alle Rendite-Risiko Kombinationen $f(a)$, die durch die Ergebnisse zulässiger Portfolios x dominiert werden, d. h. $f(a) \leq f(x)$. Auf der x -Achse befindet sich die erwartete Rendite, während auf der invertierten y -Achse das Risiko in Form der Varianz abgebildet ist.

Die Funktion $f(x)$ stellt wie zuvor in $(\text{PO}(Q, r))$ das Ergebnis der Investitionsstrategie $x \in \mathbb{R}^n$ dar, die in der ersten Stufe von den Investierenden festgelegt wird. Die Matrix $V \in \mathbb{R}^{2 \times m}$ beschreibt m verschiedene Unsicherheitsrichtungen, in die sich das Portfolio in der zweiten Stufe verschieben kann. Die Skalierungsfunktion $\delta(x)$ verstrt diese Verschiebungen in Abhigkeit der Investitionsstrategie x . Indem $\delta(x)V$ addiert wird, ergibt sich eine Punktmenge an Ergebnissen. Von dieser Punktmenge wird anschlieend die konvexe Hule gebildet. Dies entspricht dann der Menge an mglichen Ergebnissen, die durch die Wahl der Investitionsstrategie x auf zweiter Stufe nach Realisierung der Unsicherheit erreicht werden knnen. Anschlieend wird der Kegel $-\mathbb{R}_+^2$ addiert, d. h. $F(x) : \mathbb{R}^n \rightrightarrows \mathbb{R}^2$ beinhaltet nun alle mglichen Portfolioergebnisse $F(\tilde{x})$, die durch die Strategie x dominiert werden: $F(\tilde{x}) \subseteq F(x)$. Dementsprechend wird in diesem konkreten Kontext angenommen, dass die Investierenden eine grere Flexibilitt prferieren und zugleich ihre Wahl der Investitionsstrategie $x \in \mathbb{R}^n$ anhand dessen treffen, welches Portfolio unter Einbezug der Unsicherheit die bestmglichen Ergebnisse erzielen knnte. Es wird also ein Portfolio gesucht, das gleichzeitig die Rendite maximiert, Risiko minimiert und zustzlich von keinem anderen Portfolio dominiert wird.

Beispiel 5.2. Es seien $n = 2$ Assets mit der im Nachfolgenden festgelegten Rendite r , Kovarianzmatrix Q und Unsicherheitsrichtungen V gegeben.

$$x = \begin{pmatrix} x_1 \\ 1 - x_1 \end{pmatrix} \quad r = \begin{pmatrix} 0.095 \\ 0.105 \end{pmatrix} \quad Q = \begin{pmatrix} 0.030 & -0.012 \\ -0.012 & 0.020 \end{pmatrix}$$

$$V = \begin{pmatrix} 0.010 & 0.020 & -0.006 & 0.000 \\ -0.005 & 0.015 & 0.012 & -0.008 \end{pmatrix}$$

Zudem sei $\delta(x) = \frac{3}{4}x_1(1 - x_1)$. D. h. Diversifizierung wird belohnt, da die Menge, die von der konvexen Hule in $F(x)$ erzeugt wird, durch die Skalierung mit $\delta(x)$ grer wird je nher x_1 dem Wert $\frac{1}{2}$ ist. Eine Visualisierung der Funktion $F(x)$ gemig $(\text{MPO}(Q, r, \delta, V))$ ist in Abbildung 3 dargestellt. Aus Abbildung 3 lassen sich sofort zwei Strategien ablesen, die zwar beide pareto-optimal sind, aber eine Strategie die andere dominiert: fr $x_a = (0, 1)^T$ und $x_b = (\frac{1}{5}, \frac{4}{5})^T$ ist $F(x_a) \subseteq F(x_b)$.

Ausgehend von dieser Observation werden dem implementierten Algorithmus 2 die angegebenen Werte fr $r, Q, V, \delta(x)$, das Startportfolio $x_0 = (0, 1)^T$ und die Funktion F bergeben. Algorithmus 2 identifiziert nach einer Iteration den korrekten Optimierer $\bar{x} = (\frac{1}{3}, \frac{2}{3})^T$. Der Losungsprozess wird in Abbildung 4 visualisiert.

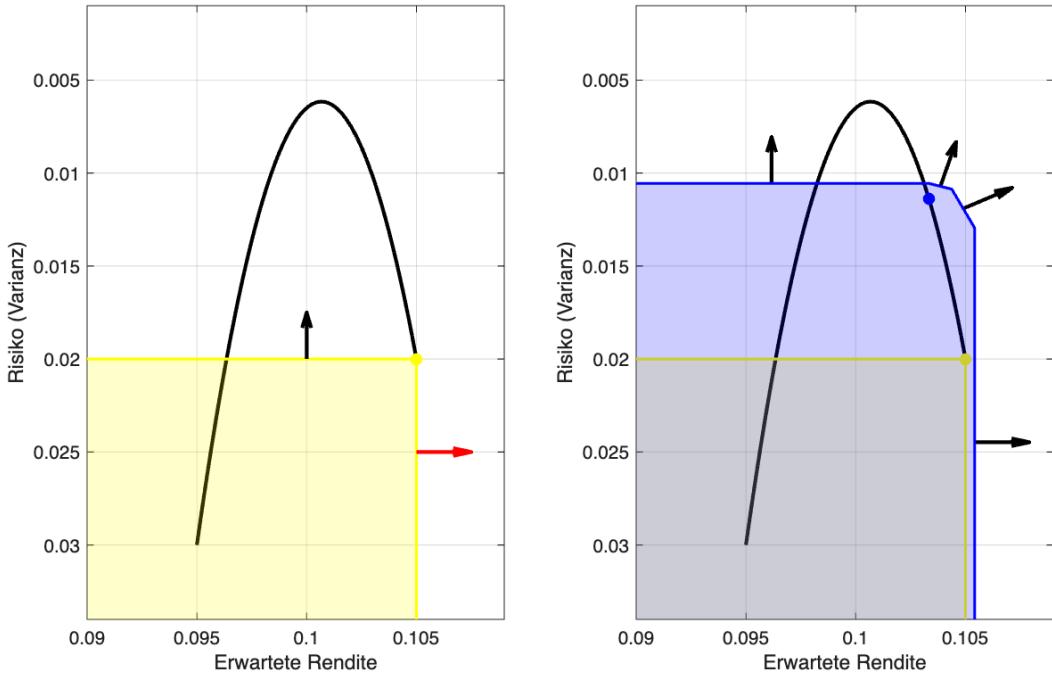


Abbildung 4: In beiden Grafiken ist $F(x_0)$ für das Startportfolio $x_0 = (0, 1)^T$ in gelb abgebildet. Der gelbe Punkt ist hierbei $f(x_0)$. Der schwarze und rote Pfeil im linken Bild stellen die relevanten Außennormalen von $F(x_0)$ dar, die der Algorithmus 2 testet. Der Algorithmus 2 wählt den ersten Verbesserungsschritt in Richtung des roten Pfeils. In der rechten Grafik ist der Funktionswert der Strategie $\bar{x} = (\frac{1}{3}, \frac{2}{3})^T$, die die maximale Verbesserung in die rote Richtung ermöglicht, in blau dargestellt. $f(\bar{x})$ ist erneut als Punkt in derselben Farbe hervorgehoben. Die Außennormalen von $F(\bar{x})$ sind als schwarze Pfeile abgebildet: in diese Richtungen ist keine strikte Verbesserung mehr möglich. Die schwarze Kurve beschreibt jeweils $f(x)$ für alle zulässigen Strategien x . Es sind die Werte aus Beispiel 5.2 verwendet worden und die Achsen sind die selben wie in Abbildung 3: erwartete Rendite auf der x -Achse und Varianz auf der y -Achse.

5.3 Implementierungsdetails

Die beiden Beispiele 5.1 und 5.2 sowie der Algorithmus 2 des vorherigen Abschnitts 4 sind in MATLAB (R2025b) Version 25.2 [31] unter Verwendung von CVX [1, 6, 11] implementiert worden. Jede Berechnung wurde auf einem Apple MacBook Pro (2022) mit Apple M2 Prozessor durchgeführt. Der gesamte, vollständig auskommentierte Code mit den konkreten Implementierungen des Algorithmus 2 für die Beispiele 5.1 und 5.2 sind verfügbar auf <https://github.com/SarahStudent/Masterarbeit/tree/main>.

In diesem Abschnitt wird auf spezifische Implementierungsdetails eingegangen. Das Beispiel 5.1 lässt sich weitestgehend analog zu Algorithmus 2 implementieren. Bei den Problemen $(\text{CP}(F, w, I_{F(\bar{x}}))$ und $(\text{IP}(F, \bar{x}, p, d))$) werden für die Restriktionen $y \in F(x)$ und $I \subseteq F(x)$ jeweils die Ausdrücke (6) und (7) eingesetzt.

$$y \in F(x) \quad \hat{=} \quad \begin{cases} 0 \leq x; \\ x \leq 2; \\ \text{norm}(y - [0; 0], 2) \leq 1 + 0.3 * x; \end{cases} \quad (6)$$

$$I \subseteq F(x) \quad \hat{=} \quad \begin{cases} \text{for } j = 1 : \text{size}(I, 2) \\ \quad \text{norm}(I(:, j) - [0; 0], 2) \leq 1 + 0.3 * x; \\ \text{end} \end{cases} \quad (7)$$

Beim zweiten Beispiel 5.2 ergeben sich bei der Verwendung von CVX [1, 6, 11] teilweise Schwierigkeiten, die Restriktionen $y \in F(x)$ und $I \subseteq F(x)$ umzusetzen. Aus diesem Grund wird hier auf explizite geometrische Eigenschaften zurückgegriffen. Bei der Menge $F(x)$ handelt es sich um einen Polyeder, sodass anstelle der Innenapproximation I die Menge selbst verwendet werden kann. Des Weiteren kann in der Implementierung der Polyeder $F(x)$ außerhalb des relevanten Bereichs abgeschnitten werden, da die Richtungen $(-1, 0)^T$ (weniger Rendite) und $(0, 1)^T$ (höheres Risiko) im betrachteten Kontext der Portfoliooptimierung nicht relevant sind. D. h. es wird nicht der gesamte Kegel $-\mathbb{R}_+^2$ verwendet, sondern lediglich eine repräsentative, endliche Menge. Dadurch stellt $F(x)$ in der Implementierung ein Polytop dar und die Menge der bereits berücksichtigten Richtungen wird mit $W = \{(-1, 0)^T, (0, 1)^T\}$ initialisiert.

Da es sich bei $F(x)$ um ein Polytop handelt, ist es ausreichend, für eine feste Richtung $w \in N \setminus W$ die lineare Zielfunktion $f(y) = w^T y$ aus $(\text{CP}(F, w, I_{F(\bar{x}}))$ bezüglich der Eckpunkte des Polytops zu maximieren, weil lineare Funktionen auf Polytopen ihr Optimum stets in einem Extrempunkt annehmen [vgl. 24, S. 54f]. Um ein strikt besseres Portfolio zu finden, werden für die Richtung w und alle Eckpunkte y die maximalen Werte der Zielfunktion $f(y) = w^T y$ bestimmt. Dabei werden ausschließlich die zulässigen Portfolios betrachtet, die aufgrund von $n = 2$ in diesem Beispiel eine Kurve bilden und eindeutig durch $x_1 \in [0, 1]$ beschrieben werden können. Aufgrund der Konvexität und Kompaktheit der betrachteten Mengen sowie der Linearität der Funktion $f(y) = w^T y$ werden alle relevanten Extremwerte der Zielfunktion erfasst. Da die Ausgangsmenge $F(x_0)$ rechteckig ist und die optimale Strategie bereits nach der ersten Iteration identifiziert wird, kann die Inklusionsprüfung $F(x_0) \subseteq F(\bar{x})$ in der Implementierung approximativ mittels einer koordinatenweisen Abschätzung durch Minimal- und Maximalwerte umgesetzt werden.

6 Diskussion

In der vorliegenden Arbeit wurde ein korrekter und endlicher Algorithmus zur Lösung bestimmter kompakt-konvexer Mengenoptimierungsprobleme unter Verwendung von ϵ -Optimierern entwickelt. Zwei Beispielprobleme wurden mithilfe einer Implementierung des Algorithmus 2 gelöst. Die Ergebnisse sowie der Lösungsprozess wurden anschließend illustriert.

Beide Beispiele sind aus Visualisierungsgründen einfach gehalten, weshalb teilweise bereits nach einer Iteration ein ϵ -Optimierer identifiziert wird. Das zweite Beispiel zur Portfoliooptimierung orientiert sich zwar inhaltlich an der Praxis, stellt allerdings ein stark vereinfachtes Modell dar: es werden lediglich zwei Assets betrachtet und die Unsicherheiten sind vorher bekannt. Dementsprechend dient dieses Beispiel im Rahmen der Arbeit lediglich dazu, einen ersten Eindruck zu vermitteln und einen weiterführenden Ausblick darauf zu geben, in welchen praxisrelevanten Kontexten der hier entwickelte Lösungsalgorithmus eingesetzt werden könnte.

Darüber hinaus baut der implementierte Algorithmus teilweise stark auf der jeweiligen Struktur der Beispiele auf. Während im ersten Beispiel eine Implementierung der Restriktionen direkt im Algorithmus möglich ist, wird im zweiten Beispiel auf individuelle geometrische Eigenschaften des verwendeten Problems zurückgegriffen, die im Allgemeinen auf konvexe Mengen nicht zutreffen. Vorteilhafter wäre eine allgemeinere und wiederverwendbare Implementierung, bei der eine beliebige mengenwertige Funktion $F(x)$ an den Algorithmus übergeben werden kann, z. B. in Form einer von x abhängigen, konvexen Menge.

Des Weiteren ist der entwickelte Algorithmus auf kompakt-konvexe Mengenoptimierungsprobleme beschränkt. Obwohl die Kompaktheit für die Korrektheit und die Endlichkeit des Algorithmus von zentraler Bedeutung ist, bleibt mit dieser Einschränkung ein wesentlicher Teil konvexer Probleme unberücksichtigt. Allerdings stellt der hier entwickelte Algorithmus einen der ersten Ansätze zur Lösung konvexer Mengenoptimierungsprobleme dieser Art dar und bildet damit einen wichtigen Ausgangspunkt für weiterführende Forschung, insbesondere im Hinblick auf eine mögliche Erweiterung des Ansatzes auf nichtkompakte, konvexe Probleme.

7 Fazit

In dieser Arbeit werden ϵ -Optimierer als Lösungsansatz für kompakt-konvexe Mengenoptimierungsprobleme eingeführt und deren Existenz nachgewiesen. Des Weiteren wird ein Algorithmus zur Berechnung von ϵ -Optimierern entwickelt. Die Korrektheit des Algorithmus wird ohne weitere Voraussetzungen bewiesen, während seine Endlichkeit unter der zusätzlichen Bedingung gilt, dass der Ursprung im Inneren des Graphen der Zielfunktion liegt. Anschließend werden zwei beispielhafte Probleme mithilfe einer Implementierung des Algorithmus gelöst.

Literaturverzeichnis

- [1] M. Adamaszek und W. Wang. *CVX for Apple Silicon*. An unofficial native CVX 2.2 package for apple silicon. 2024. URL: <https://ask.cvxr.com/t/announcement-cvx-for-apple-silicon/12280> (besucht am 17. 11. 2025).
- [2] W. Alt. *Numerische Verfahren der konvexen, nichtglatten Optimierung*. Teubner Verlag, Wiesbaden, 2004.
- [3] J. Aubin und H. Frankowska. *Set-valued analysis*. Birkhäuser, Boston, 1990.
- [4] T. Bao und B. Mordukhovich. “Set-valued optimization in welfare economics”. In: *Advances in Mathematical Economics*. Hrsg. von S. Kusuoka und T. Maruyama. Springer Japan, Tokyo, 2010, S. 113–153.
- [5] S. Boyd und L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.
- [6] Inc. CVX Research. *CVX: Matlab software for disciplined convex programming, version 2.0*. <https://cvxr.com/cvx>. 2012.
- [7] D. Dörfler. “Calculus of unbounded spectrahedral shadows and their polyhedral approximation”. Diss. Friedrich-Schiller-Universität Jena, 2023.
- [8] Dudenredaktion, Hrsg. *Duden Die deutsche Rechtschreibung*. 27. Aufl. Dudenverlag, 2017.
- [9] J. Elstrodt. *Maß- und Integrationstheorie*. 7. Aufl. Springer Berlin, Heidelberg, 2011.
- [10] O. Forster. *Analysis 2*. 8. Aufl. Vieweg+Teubner, Wiesbaden, 2008.
- [11] M. Grant und S. Boyd. “Graph implementations for nonsmooth convex programs”. In: *Recent advances in learning and control*. Hrsg. von V. Blondel, S. Boyd und H. Kimura. Lecture notes in control and information sciences. http://stanford.edu/~boyd/graph_dcp.html. Springer-Verlag Limited, 2008, S. 95–110.
- [12] A. Hamel und A. Löhne. “Multi-objective stochastic linear programming with recourse and flexible decision making”. In: (2025). arXiv: 2407 . 04602 [math.OC]. URL: <https://arxiv.org/abs/2407.04602> (besucht am 19. 11. 2025).
- [13] A. Hamel u. a. *Set optimization and applications - the state of the art*. Springer Berlin, Heidelberg, 2015.
- [14] F. Hausdorff u. a. *Felix Hausdorff - Gesammelte Werke Band II*. Springer Berlin, Heidelberg, 2002.

- [15] E. Hernández und L. Rodríguez-Marín. "Lagrangian duality in set-valued optimization". In: *Journal of Optimization Theory and Applications* 134 (2007), S. 119–134.
- [16] J. Jahn. "Grundlagen der Mengenoptimierung". In: *Multi-Criteria- und Fuzzy-Systeme in Theorie und Praxis: Lösungsansätze für Entscheidungsprobleme mit komplexen Zielsystemen*. Hrsg. von W. Habenicht, B. Scheubrein und R. Scheubrein. Deutscher Universitätsverlag, 2003, S. 37–71.
- [17] K. Jänich. *Topologie*. 8. Aufl. Springer Berlin, Heidelberg, 2008.
- [18] J. Jantzen und J. Schwermer. *Algebra*. Springer Berlin, Heidelberg, 2006.
- [19] G. Kamenev. "A class of adaptive algorithms for the approximation of convex bodies by polyhedra". In: *Zh. Vychisl. Mat. i Mat. Fiz* 32.1 (1992), S. 136–152.
- [20] G. Kamenev. "Analysis of an algorithm for approximating convex bodies". In: *Zh. Vychisl. Mat. i Mat. Fiz* 34.4 (1994), S. 608–616.
- [21] C. Karpfinger und K. Meyberg. *Algebra*. 2. Aufl. Spektrum Akad.-Verl., Heidelberg, 2010.
- [22] A. Khan, C. Tammer und C. Zălinescu. *Set-valued optimization*. Springer Berlin, Heidelberg, 2015.
- [23] K. Königsberger. *Analysis* 2. 5. Aufl. Springer Berlin, Heidelberg, 2004.
- [24] A. Koop und H. Moock. *Lineare Optimierung: eine anwendungsorientierte Einführung in Operations Research*. Spektrum Akad.-Verl., Heidelberg, 2008.
- [25] D. Kreps. "A representation theorem for "preference for flexibility"". In: *Econometrica* 47.3 (1979), S. 565–577.
- [26] D. Lau. *Algebra und diskrete Mathematik 1*. 2. Aufl. Springer Berlin, Heidelberg, 2007.
- [27] S. Lay. *Convex sets and their applications*. Krieger, Malabar, 1992.
- [28] A. Löhne. "A decision-making method in polyhedral convex set optimization". In: (2024). arXiv: 2409.17998 [math.OC]. URL: <https://arxiv.org/abs/2409.17998> (besucht am 09.12.2025).
- [29] A. Löhne. "A solution method for arbitrary polyhedral convex set optimization problems". In: *SIAM Journal on Optimization* 35.1 (2025), S. 330–346.
- [30] A. Löhne. *Vector optimization with infimum and supremum*. Springer Berlin, Heidelberg, 2011.
- [31] MATLAB. *Version 25.2.0.3042426 (R2025b)*. Natick, Massachusetts, 2025. URL: <https://www.mathworks.com> (besucht am 10.11.2025).

- [32] E. Ossa. *Topologie : eine anschauliche Einführung in die geometrischen und algebraischen Grundlagen*. 2. Aufl. Vieweg+Teubner, Wiesbaden, 2009.
- [33] *pymoo - portfolio allocation*. URL: https://pymoo.org/case_studies/portfolio_allocation.html (besucht am 11. 11. 2025).
- [34] L. Richardson. *Advanced calculus*. John Wiley & Sons, Hoboken, 2008.
- [35] R. Rockafellar. *Convex analysis*. 2. Aufl. Princeton University Press, 1972.
- [36] R. Rockafellar und R. Wets. *Variational analysis*. Springer Berlin, Heidelberg, 1998.
- [37] L. Salomon, D. Dörfler und A. Löhne. “MOCVXPY: a CVXPY extension for multiobjective optimization”. In: (2025). arXiv: 2510.21010 [math.OC]. URL: <https://arxiv.org/abs/2510.21010> (besucht am 10. 11. 2025).
- [38] G. Ziegler. *Lectures on polytopes*. 2. Aufl. Springer New York, 1998.

Verzeichnis der Algorithmen

1	Kernalgorithmus zur Berechnung eines Optimierers für polyedrisch-konvexe Probleme	11
2	Algorithmus zur Berechnung eines ϵ -Optimierers für kompakt-konvexe Probleme	12
3	Funktion zur Berechnung einer initialen Innenapproximation I eines kompakt-konvexen Körpers $F(\bar{x})$	16
4	Funktion zur Aktualisierung der Innenapproximation I eines kompakt-konvexen Körpers $F(\bar{x})$ mit $y^* \in \text{bd } F(\bar{x})$	16

Abbildungsverzeichnis

1	Eine Illustration der Algorithmen 2, 3 und 4 anhand des Beispiels 5.1 .	18
2	Von Algorithmus 2 berechnete Approximationen für Beispiel 5.1 mit $\epsilon \in \{0.1, 0.01\}$	19
3	$F(x)$ für bestimmte $x \in \mathbb{R}^2$ mit Werten aus Beispiel 5.2	20
4	Lösung von Algorithmus 2 für $F(x)$ aus Beispiel 5.2 mit Startportfolio $x = (0, 1)^T$	22

Eigenständigkeitserklärung

1. Hiermit versichere ich, dass ich die vorliegende Arbeit - bei einer Gruppenarbeit die von mir zu verantwortenden und entsprechend gekennzeichneten Teile - selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich trage die Verantwortung für die Qualität des Textes sowie die Auswahl aller Inhalte und habe sichergestellt, dass Informationen und Argumente mit geeigneten wissenschaftlichen Quellen belegt bzw. gestützt werden. Die aus fremden oder auch eigenen, älteren Quellen wörtlich oder sinngemäß übernommenen Textstellen, Gedankengänge, Konzepte, Grafiken etc. in meinen Ausführungen habe ich als solche eindeutig gekennzeichnet und mit vollständigen Verweisen auf die jeweilige Quelle versehen. Alle weiteren Inhalte dieser Arbeit ohne entsprechende Verweise stammen im urheberrechtlichen Sinn von mir.
2. Ich weiß, dass meine Eigenständigkeitserklärung sich auch auf nicht zitierfähige, generierende KI- Anwendungen (nachfolgend „generierende KI“) bezieht. Mir ist bewusst, dass die Verwendung von generierender KI unzulässig ist, sofern nicht deren Nutzung von der prüfenden Person ausdrücklich freigegeben wurde (Freigabeerklärung). Sofern eine Zulassung als Hilfsmittel erfolgt ist, versichere ich, dass ich mich generierender KI lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss deutlich überwiegt. Ich verantworte die Übernahme der von mir verwendeten maschinell generierten Passagen in meiner Arbeit vollumfänglich selbst. Für den Fall der Freigabe der Verwendung von generierender KI für die Erstellung der vorliegenden Arbeit wird eine Verwendung in einem gesonderten Anhang meiner Arbeit kenntlich gemacht. Dieser Anhang enthält eine Angabe oder eine detaillierte Dokumentation über die Verwendung generierender KI gemäß den Vorgaben in der Freigabeerklärung der prüfenden Person. Die Details zum Gebrauch generierender KI bei der Erstellung der vorliegenden Arbeit inklusive Art, Ziel und Umfang der Verwendung sowie die Art der Nachweispflicht habe ich der Freigabeerklärung der prüfenden Person entnommen.
3. Ich versichere des Weiteren, dass die vorliegende Arbeit bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt wurde oder in deutscher oder einer anderen Sprache als Veröffentlichung erschienen ist.

4. Mir ist bekannt, dass ein Verstoß gegen die vorbenannten Punkte prüfungsrechtliche Konsequenzen haben und insbesondere dazu führen kann, dass meine Prüfungsleistung als Täuschung und damit als mit „nicht bestanden“ bewertet werden kann. Bei mehrfachem oder schwerwiegendem Täuschungsversuch kann ich befristet oder sogar dauerhaft von der Erbringung weiterer Prüfungsleistungen in meinem Studiengang ausgeschlossen werden.

Jena, den 28. Januar 2026
