

Rumpfskript zur Vorlesung
**Einführung in die Mathematik
des Operations Research**
Sommersemester 2018

Prof. Dr. Frank Vallentin

Mathematisches Institut
Universität zu Köln
Weyertal 86–90
50931 Köln
Germany

Inhaltsverzeichnis

Teil A. Graphen und Netzwerke, kombinatorische Algorithmen	5
Kapitel 1. Kürzeste Wege	7
1. Gerichtete Graphen	7
2. Berechnung kürzester Wege	8
3. Kürzeste Wege und ganzzahlige lineare Optimierung	8
4. Potentiale	9
Kapitel 2. Matchings in bipartiten Graphen	11
1. Grundbegriffe	11
2. Berechnung von Matchings mit maximaler Kardinalität	11
3. Das Matchingtheorem von König	12
4. Die ungarische Methode	13
Kapitel 3. Flüsse in Netzwerken	15
1. Das Max-Flow-Min-Cut-Theorem	15
Teil B. Einführung in die konvexe und lineare Optimierung	19
Kapitel 4. Konvexität	21
1. Grundbegriffe aus Topologie und Geometrie	21
2. Konvexe Mengen und konvexe Funktionen	22
3. Trennungssätze	24
4. Repräsentation konvexer Mengen	25
Kapitel 5. Polyedertheorie	27
1. Polyeder und Polytope	27
2. Das Lemma von Farkas	28
3. Lineare Optimierung / Lineare Programmierung	28
4. Das Eliminationsverfahren von Fourier und Motzkin	29
Kapitel 6. Ganzzahlige lineare Optimierung und vollständig unimodulare Matrizen	33
1. Ganzzahlige lineare Programme	33
2. Vollständig unimodulare Matrizen	33
3. Vollständig-unimodulare Matrizen und bipartite Graphen	34
4. Vollständig-unimodulare Matrizen und gerichtete Graphen	34
5. Charakterisierung von vollständig unimodularen Matrizen	35
Teil C. Algorithmen der linearen Optimierung	37
Kapitel 7. Das Simplexverfahren	39
1. Simplexalgorithmus mit bekannter Startecke	39
2. Simplexalgorithmus ohne Startecke	41
3. Zur praktischen und theoretischen Effizienz des Simplexalgorithmus	42

Kapitel 8. Die Ellipsoidmethode	43
1. Grundlagen zu Ellipsoiden	43
2. Trennen und Optimieren	44
3. Ellipsoidmethode	46

Teil A

**Graphen und Netzwerke,
kombinatorische Algorithmen**

KAPITEL 1

Kürzeste Wege

1. Gerichtete Graphen

DEFINITION 1.1. Ein gerichteter Graph (Netzwerk) $D = (V, A)$ ist ein Paar, bestehend aus einer endlichen Menge V , die Menge der Knoten von D , und einer Menge A mit

$$A \subseteq \{(v, w) \in V \times V : v \neq w\},$$

die Menge der gerichteten Kanten von D . Mit anderen Worten: Ein gerichteter Graph beschreibt eine irreflexive Relation auf $V \times V$.

DEFINITION 1.2. Eine Kantenfolge P in D ist ein Tupel der Form

$$P = (v_0, a_1, v_1, a_2, v_2, \dots, a_m, v_m)$$

mit den Eigenschaften $v_0, \dots, v_m \in V$, $a_1, \dots, a_m \in A$ und $a_i = (v_{i-1}, v_i)$ für $i = 1, \dots, m$. Der Knoten v_0 heißt Startknoten von P und der Knoten v_m heißt Endknoten von P . Wir sprechen auch von einer Kantenfolge von v_0 nach v_m . Eine Kantenfolge P heißt ein Weg von v_0 nach v_m oder kurz Weg, falls

$$|\{v_0, \dots, v_m\}| = m + 1$$

gilt, d.h. die in P auftretenden Knoten sind alle paarweise verschieden. Eine Kantenfolge P heißt (gerichteter) Kreis, falls $v_0 = v_m$ und falls

$$|\{v_0, \dots, v_m\}| = m$$

gilt.

DEFINITION 1.3. Sei $D = (V, A)$ ein gerichteter Graph. Auf den Kanten definieren wir Kantenlängen durch eine Kantenlängenfunktion $l \in \mathbb{R}^A$ wobei es durchaus vorkommen kann, dass Kanten eine negative Länge besitzen. Die Länge einer Kantenfolge P definieren wir als

$$l(P) = \sum_{i=1}^m l(a_i).$$

Für zwei Knoten s und t in V definieren wir den Abstand (Distanz) als

$$\text{dist}(s, t) = \inf\{l(P) : P \text{ ist ein Weg von } s \text{ nach } t\}.$$

Ist eine Kantenfolge P ein Weg von s nach t mit $l(P) = \text{dist}(s, t)$, dann heißt P ein kürzester Weg von s nach t .

SATZ 1.4. Es sei $D = (V, A)$ ein gerichteter Graph mit Längenfunktion $l \in \mathbb{R}^A$. Angenommen alle gerichteten Kreise in D haben nicht-negative Länge und angenommen es gibt einen Weg von s nach t , d.h. $\text{dist}(s, t) < \infty$. Dann existiert eine kürzeste Kantenfolge mit Startknoten s und Endknoten t , die ein Weg ist.

Input : Gerichteter Graph $D = (V, A)$, $n = |V|$, $s \in V$, $l \in \mathbb{R}^A$
Output : Funktionen $d_0, \dots, d_n \in (\mathbb{R} \cup \{\infty\})^V$, $g: V \setminus \{s\} \rightarrow V$
 Setze $d_0(s) = 0$.
 Setze $d_0(v) = \infty \forall v \in V \setminus \{s\}$.
for $k = 0$ **to** $n - 1$ **do**
 $d_{k+1} = d_k$
 for alle Kanten $(u, v) \in A$ **do**
 if $d_{k+1}(v) > d_k(u) + l(u, v)$ **then**
 $d_{k+1}(v) = d_k(u) + l(u, v)$
 $g(v) = u$
 end
 end
end
if $d_n \neq d_{n-1}$ **then**
 Ausgabe: Es gibt Kreis negativer Länge, der von s aus erreichbar ist.
end

Algorithmus 1 : Algorithmus von Bellman und Ford

2. Berechnung kürzester Wege

SATZ 2.1. Nach Ablauf der k -ten Iteration der äußeren for-Schleife ($k = 0, \dots, n - 1$) im Algorithmus von Bellman und Ford gilt

$$d_{k+1}(v) = \inf\{l(P) : P \text{ ist Kantenfolge von } s \text{ nach } v \text{ mit höchstens } k + 1 \text{ Kanten}\}.$$

SATZ 2.2. Nach Ablauf des Algorithmus von Bellman und Ford gilt $d_n = d_{n-1}$ genau dann, wenn alle von s aus erreichbaren Kreise nicht-negative Länge haben.

BEMERKUNG 2.3. (a) Die Laufzeit des Algorithmus von Bellman und Ford ist proportional zu $|V| \cdot |A| \leq |V|^3$.

(b) Falls alle von s aus erreichbaren Kreise nicht-negative Länge besitzen, dann ist $\text{dist}(s, v) = d_{n-1}(v)$ für alle $v \in V$. Dann ist auch

$$v, g(v), g(g(v)), \dots, s$$

die Umkehrung eines kürzesten Weges von s nach v .

3. Kürzeste Wege und ganzzahlige lineare Optimierung

DEFINITION 3.1. Seien $c \in \mathbb{Z}^n$, $b \in \mathbb{Z}_{\geq 0}^m$, $M \in \mathbb{Z}_{\geq 0}^{m \times n}$ gegeben. Ein ganzzahliges lineares Optimierungsproblem (ILP = integer linear program) ist von der Form

$$\max\{c^\top x : x \in \mathbb{Z}^n, x_i \geq 0 \ (i = 1, \dots, n), (Mx)_j \leq b_j \ (j = 1, \dots, m)\}.$$

Definiere den gerichteten Graph $D = (V, A)$ mit Knoten

$$V = \{0, \dots, n\} \times U \quad \text{mit} \quad U = \{0, \dots, b_1\} \times \dots \times \{0, \dots, b_m\}$$

und Kanten für $i = 1, \dots, n$ und $u \in U$

$$((i - 1, u), (i, u + km_i)) \in A$$

mit $k \in \mathbb{Z}_{\geq 0}$, falls $u + km_i \in U$, wobei m_i die i -te Spalte der Matrix M ist. Definiere die Kantenlängenfunktion $l \in \mathbb{R}^A$ durch

$$l(((i - 1, u), (i, u + km_i))) = -c_i k.$$

Nun entsprechen kürzeste Wege von $(0, (0, \dots, 0))$ nach (n, u) mit $u \in U$ genau denen $x \in \mathbb{Z}^n$ mit $x_i \geq 0$ für $i = 1, \dots, n$ und $(Mx)_j \leq b_j$ für $j = 1, \dots, m$ mit maximalem Wert $c^\top x$.

4. Potentiale

Wie kann man beweisen, dass ein gegebener Weg von s nach t tatsächlich ein kürzester Weg von s nach t ist? Eine mögliche Beweisstrategie ist es, einfach alle Wege aufzuzählen. Diese Strategie wird auch durch die Definition eines kürzesten Weges nahegelegt. Wir haben jedoch schon gesehen, dass es exponentiell viele Wege von s nach t geben kann, so dass dieses vollständige Aufzählen im Allgemeinen sehr ineffizient ist. Das Ziel dieses Unterkapitels ist es, eine alternative, viel effizientere Beweismethode anzugeben.

DEFINITION 4.1. Sei $D = (V, A)$ ein gerichteter Graph mit Kantenlängenfunktion $l \in \mathbb{R}^A$. Eine Funktion $p \in \mathbb{R}^V$ heißt Potential für D und l , falls die Ungleichung

$$p(v) - p(u) \leq l(a) \quad \text{für alle Kanten } a = (u, v) \in A.$$

gilt.

SATZ 4.2. Sei $D = (V, A)$ ein gerichteter Graph mit Kantenlängenfunktion $l \in \mathbb{R}^A$. Es existiert genau dann ein Potential $p \in \mathbb{R}^V$ für D und l , wenn alle gerichteten Kreise in D eine nicht-negative Länge besitzen.

Es bleibt auch zu bemerken, dass $p = 0$ ein Potential ist, falls die Kantenlängenfunktion nur nicht-negative Werte annimmt.

SATZ 4.3. (geometrische Modellierung kürzester Wege mit linearen Ungleichungen, Min-Max-Charakterisierung)

Es sei $D = (V, A)$ ein gerichteter Graph und $l \in \mathbb{R}^A$ eine Kantenlängenfunktion. Angenommen alle gerichteten Kreise in D und l haben nicht-negative Länge. Sei $s \in V$ und $t \in V$ Knoten und angenommen für alle Knoten $v \in V$ gilt $\text{dist}(s, v) < \infty$. Dann gilt die folgende Min-Max-Charakterisierung für die Länge eines kürzesten Weges von s nach t :

$$\begin{aligned} \text{dist}(s, t) &= \min\{l(P) : P \text{ ist Weg von } s \text{ nach } t\} \\ &= \max\{p(t) - p(s) : p \in \mathbb{R}^V \text{ Potential von } D \text{ und } l\}. \end{aligned}$$

Optimale Potentiale, d.h. Potentiale an denen das Maximum $p(t) - p(s)$ angenommen wird, sind im übrigen nicht eindeutig bestimmt. Zum Beispiel ist, wenn p ein Potential ist, auch $p + C$ für jede Konstante $C \in \mathbb{R}$ ein Potential.

Matchings in bipartiten Graphen

1. Grundbegriffe

DEFINITION 1.1. Ein (ungerichteter) Graph $G = (V, E)$ ist ein Paar, bestehend aus einer endlichen Menge V , Menge der Knoten (vertices), und einer Menge $E \subseteq \{\{v, w\} : v, w \in V, v \neq w\}$, Menge der Kanten (edges).

Man kann die Begriffe Kantenfolge, Weg und Kreis für ungerichtete Graphen genauso wie bei gerichteten Graphen definieren.

DEFINITION 1.2. Sei $G = (V, E)$ in Graph.

- (1) Zwei Knoten v, w heißen benachbart (adjazent), wenn $\{v, w\} \in E$. Ein Knoten $v \in V$ heißt inzident zu einer Kante $e \in E$, falls $v \in e$.
- (2) Zwei Knoten $v, w \in V$ heißen wegzusammenhängend, falls es in G einen Weg von v nach w gibt.
- (3) Dies definiert eine Äquivalenzrelation auf V und ihre Äquivalenzklassen heißen Zusammenhangskomponenten.
- (4) Falls G nur eine Zusammenhangskomponente besitzt, so heißt G zusammenhängend.

DEFINITION 1.3. Ein ungerichteter Graph $G = (V, E)$ heißt bipartit, falls es Teilmengen $U, W \subseteq V$ gibt, so dass

- (1) die Mengen U und W eine Partition (eine Bipartition) von V sind, d.h.

$$V = U \cup W, \quad U \cap W = \emptyset,$$

- (2) jede Kante von G je nur einen Knoten aus U und einen aus W besitzt:

$$|e \cap U| = |e \cap W| = 1 \quad \text{für alle } e \in E.$$

2. Berechnung von Matchings mit maximaler Kardinalität

DEFINITION 2.1. Es sei $G = (V, E)$ ein ungerichteter Graph.

- (1) Ein Matching $M \subseteq E$ in G ist eine Menge disjunkter Kanten, das heißt

$$\forall e, f \in M, e \neq f : e \cap f = \emptyset.$$

- (2) Die Matchingzahl von G ist

$$\nu(G) = \max\{|M| : M \subseteq E \text{ ist Matching in } G\}.$$

- (3) Ein Matching heißt perfekt, falls $2|M| = |V|$ gilt.

Im folgenden identifizieren wir einen Weg $P = (v_0, e_1, v_1, \dots, e_m, v_m)$ mit der Menge seiner Kanten: Wir schreiben vereinfachend $P = \{e_1, \dots, e_m\} \subseteq E$.

DEFINITION 2.2. Es sei $G = (V, E)$ ein ungerichteter Graph und es sei $M \subseteq E$ ein Matching in G . Ein Weg $P \subseteq E$ heißt M -augmentierend, falls die folgenden zwei Bedingungen erfüllt sind:

- (1) Weder Startknoten v_0 noch Endknoten v_m von P werden von M überdeckt: für alle $e \in M$ gilt $v_0 \notin e$ und $v_m \notin e$.

- (2) Die Kanten e_1, \dots, e_m von P sind alternierend nicht aus M und aus M :
 $e_1 \notin M, e_2 \in M, e_3 \notin M, \dots, e_{m-1} \in M, e_m \notin M$.

Es sei $G = (V, E)$ ein ungerichteter Graph und es sei $M \subseteq E$ ein Matching in G . Sei $P \subseteq E$ ein M -augmentierender Weg. Dann gilt:

- (1) $|P|$ ist ungerade.
 (2) $M' = M \Delta P = (M \setminus P) \cup (P \setminus M)$ ist ein Matching in M mit $|M'| = |M| + 1$.

SATZ 2.3. Es sei $G = (V, E)$ ein ungerichteter Graph und es sei $M \subseteq E$ ein Matching in G . Dann gilt entweder

- (a) $\nu(G) = |M|$,

oder

- (b) es gibt einen M -augmentierenden Weg.

Dieser Satz beweist die Korrektheit des nachfolgenden Algorithmus.

Input : Ungerichteter Graph $G = (V, E)$
Output : Matching $M \subseteq E$ mit $\nu(G) = |M|$
 $M = \emptyset$
while $\exists M$ -augmentierender Weg P **do**
 | $M = M \Delta P$.
end

Algorithmus 2 : Bestimmung eines Matchings mit maximaler Kardinalität

Nun stellt sich die Frage, wie man einen M -augmentierende Weg findet. Wenn man in den Beweis von Satz 2.3 schaut, dann erkennt man, dass dieser kein brauchbares Verfahren liefert: Um einen M -augmentierenden Weg zu konstruieren, verwendet man die Existenz eines Matchings M' , das mehr Kanten enthält als M . Ein solches Matching wollen wir aber gerade mit Hilfe eines M -augmentierenden Weges finden. Um aus diesem Henne-Ei-Problem zu entkommen, ist also die Entwicklung einer anderen Strategie notwendig.

An dieser Stelle ist es sinnvoll, sich zunächst nur bipartite Graphen anzuschauen. Denn es gibt dann einen sehr einfachen Algorithmus. Der Fall der allgemeinen Graphen ist deutlich schwieriger: Jack Edmonds fand den „Blüten“-Algorithmus für allgemeine Graphen, dieser wird aber nicht hier, sondern in Spezialvorlesungen oder in Büchern behandelt.

DEFINITION 2.4. Es sei $G = (V, E)$ ein bipartiter Graph mit Bipartition $V = U \cup W$ und es sei $M \subseteq E$ ein Matching in G . Definiere den gerichteten Residualgraph $D_M = (V, A_M)$ durch

$$A_M = \{(u, w) \in U \times W : \{u, w\} \in E \setminus M\} \cup \{(w, u) \in W \times U : \{u, w\} \in E \cap M\}.$$

SATZ 2.5. Sei $G = (V, E)$ ein bipartiter Graph mit Bipartition $V = U \cup W$. Seien $U_M \subseteq U$ und $W_M \subseteq W$ die Knoten, die nicht von M überdeckt werden. Dann entspricht jeder gerichtete Weg im Residualgraph D_M von einem Knoten in U_M zu einem Knoten in W_M einem M -augmentierenden Weg und umgekehrt.

3. Das Matchingtheorem von König

Ziel dieses Abschnittes ist es, eine Min-Max-Charakterisierung für die Matchingzahl $\nu(G)$ für einen bipartiten Graph G zu finden.

DEFINITION 3.1. Sei $G = (V, E)$ ein ungerichteter Graph.

- (1) Eine Teilmenge $C \subseteq V$ heißt Knotenüberdeckung von G , falls jede Kante einen Knoten aus C enthält, d.h.

$$\forall e \in E : |C \cap e| \geq 1.$$

- (2) Die Knotenüberdeckungszahl von G ist definiert als

$$\tau(G) = \min\{|C| : C \subseteq V \text{ ist Knotenüberdeckung von } G\}.$$

SATZ 3.2. (Das Matchingtheorem von D. König, 1931)

Sei $G = (V, E)$ ein bipartiter Graph. Dann ist die Matchingzahl von G gleich der Knotenüberdeckungszahl von G , d.h.

$$\nu(G) = \tau(G).$$

KOROLLAR 3.3. (P. Hall, 1935)

Sei $G = (V, E)$ ein bipartiter Graph. Sei $V = U \cup W$ eine Bipartition. Für eine Teilmenge $X \subseteq U$ sei $\Gamma(X)$ die Menge der Nachbarn von X :

$$\Gamma(X) = \{w \in W : \{x, w\} \in E, x \in X\}.$$

Dann gilt $\nu(G) = |U|$ genau dann, wenn für alle Teilmengen $X \subseteq U$ gilt

$$|\Gamma(X)| \geq |X|.$$

Wann und unter welchen Voraussetzungen besitzt ein bipartiter Graph ein perfektes Matching? Eine Antwort auf diese Frage gibt der sogenannte Heiratssatz.

KOROLLAR 3.4. („marriage theorem“, nach H. Weyl, 1949)

Sei $G = (V, E)$ ein bipartiter Graph. Sei $V = U \cup W$ eine Bipartition. Dann gibt es ein perfektes Matching in G genau dann, wenn $|U| = |W|$ ist und für jede Teilmenge $X \subseteq U$ gilt $|\Gamma(X)| \geq |X|$.

4. Die ungarische Methode

In diesem Abschnitt wollen wir Matchings in bipartiten Graphen berechnen, die maximales Gewicht besitzen.

DEFINITION 4.1. Sei $G = (V, E)$ ein Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion auf den Kanten von G .

- (1) Für $M \subseteq E$ definiere das Gewicht von M durch

$$w(M) = \sum_{e \in M} w(e).$$

- (2) Die gewichtete Matchingzahl von G und w ist

$$\nu_w(G) = \max\{w(M) : M \subseteq E \text{ Matching in } G\}.$$

Falls $w = 1$ ist, dann ist $\nu_w(G) = \nu(G)$.

DEFINITION 4.2. Sei $G = (V, E)$ ein Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion. Ein Matching $M \subseteq E$ in G heißt extrem für w , falls für alle Matchings $M' \subseteq E$ mit $|M'| = |M|$ gilt, dass $w(M') \leq w(M)$ ist.

DEFINITION 4.3. Sei $G = (V, E)$ ein Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion. Sei $M \subseteq E$ ein Matching in G . Definiere die Längenfunktion $l_M : E \rightarrow \mathbb{R}$ durch

$$l_M(e) = \begin{cases} w(e), & \text{falls } e \in M \\ -w(e), & \text{falls } e \notin M. \end{cases}$$

Für $P \subseteq E$ definiere

$$l_M(P) = \sum_{e \in P} l_M(e).$$

SATZ 4.4. Sei $G = (V, E)$ ein Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion. Sei $M \subseteq E$ ein extremes Matching für G und w und sei $P \subseteq E$ ein M -augmentierender Weg minimaler Länge (bzgl. l_M). Dann ist auch $M' = M \Delta P$ ein extremes Matching.

Algorithmus zur Bestimmung eines Matchings mit maximalem Gewicht: Die „ungarische Methode“ von Egerváry (1931).

Input : Ungerichteter Graph $G = (V, E)$, $w \in \mathbb{R}^E$
Output : $\nu_w(G)$
 $M_0 = \emptyset$
 $k = 1$
while $\exists M_{k-1}$ -augmentierender Weg **do**
 | Wähle M_{k-1} -augmentierenden Weg P mit minimaler Länge
 | $M_k = M_{k-1} \Delta P$
 | $k = k + 1$
end
output $\max\{w(M_i) : i = 0, 1, \dots, k - 1\}$.

Algorithmus 3 : Die ungarische Methode

Sei nun $G = (V, E)$ ein bipartiter Graph mit Bipartition $V = U \cup W$. Zur Bestimmung eines M_{k-1} -augmentierenden Weges mit minimaler Länge betrachten wir wieder den Residualgraph $D_M = (V, A_M)$ (nun mit $M = M_{k-1}$) mit Kanten

$$A_M = \{(u, w) \in U \times W : \{u, w\} \in E \setminus M\} \cup \{(w, u) \in W \times U : \{u, w\} \in E \cap M\}.$$

und (Vorsicht: operator overloading) mit Kantenlängenfunktion $l_M \in \mathbb{R}^{A_M}$ definiert durch

$$l_M((a, b)) = l_M(\{a, b\})$$

Nun finde einen kürzesten Weg von U_M nach W_M . Dies kann man mit dem Algorithmus von Bellman-Ford machen, weil D_M keine gerichteten Kreise negativer Länge besitzt, wie der folgende Satz zeigt:

SATZ 4.5. Sei nun $G = (V, E)$ ein bipartiter Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion. Sei $M \subseteq E$ ein extremes Matching. Dann besitzt der Residualgraph D_M mit Kantenlängenfunktion l_M keine gerichteten Kreise negativer Länge.

Flüsse in Netzwerken

1. Das Max-Flow-Min-Cut-Theorem

DEFINITION 1.1. *Es sei $D = (V, A)$ ein gerichteter Graph. Seien $s, t \in V$ zwei Knoten. Wir nennen s Quelle (engl. source) und t Senke (engl. terminal).*

- (1) *Eine Funktion $f \in \mathbb{R}_{\geq 0}^A$ heißt ein s - t -Fluss, falls das Flusserhaltungsgesetz*

$$\sum_{a \in \delta^{in}(v)} f(a) = \sum_{a \in \delta^{out}(v)} f(a)$$

für alle Knoten $v \in V \setminus \{s, t\}$ erfüllt ist, wobei

$$\delta^{in}(v) = \{(w, v) \in A : w \in V\},$$

$$\delta^{out}(v) = \{(v, w) \in A : w \in V\}.$$

- (2) *Der Wert eines s - t -Flusses ist definiert als*

$$\text{value}(f) = \sum_{a \in \delta^{out}(s)} f(a) - \sum_{a \in \delta^{in}(s)} f(a).$$

- (3) *Ein s - t -Fluss f heißt beschränkt durch eine Kapazitätsfunktion $c \in \mathbb{R}_{\geq 0}^A$, falls $f(a) \leq c(a)$ für alle Kanten $a \in A$ gilt. Notation: $f \leq c$.*
 (4) *Das Problem des Finden eines maximalen s - t -Flusses (Max-Flow-Problem) ist wie folgt definiert: Gegeben sei ein gerichteter Graph $D = (V, A)$, eine Quelle $s \in V$, eine Senke $t \in V$ und eine Kapazitätsfunktion $c \in \mathbb{R}_{\geq 0}^A$ und gesucht ist die Lösung des Maximierungsproblems*

$$\max \left\{ \text{value}(f) : f \in \mathbb{R}_{\geq 0}^A \text{ ist } s\text{-}t\text{-Fluss, } f \leq c \right\}.$$

Wie schon in den vorhergehenden Kapiteln wird eine Min-Max-Charakterisierung nützlich sein. Das zum Max-Flow-Problem zugehörige Minimierungsproblem ist das Min-Cut-Problem, was wir nun definieren.

DEFINITION 1.2. *Es sei $D = (V, A)$ ein gerichteter Graph mit einer Quelle $s \in V$ und einer Senke $t \in V$.*

- (1) *Ein Menge $U \subseteq V$ definiert die Schnitte*

$$\delta^{in}(U) = \{(v, u) \in A : v \in V \setminus U, u \in U\},$$

$$\delta^{out}(U) = \{(u, v) \in A : u \in U, v \in V \setminus U\}.$$

- (2) *Falls $s \in U$ und $t \notin U$ ist, dann heißen $\delta^{in}(U)$ und $\delta^{out}(U)$ s - t -Schnitte.*
 (3) *Es sei $c \in \mathbb{R}_{\geq 0}^A$ eine Kapazitätsfunktion. Dann ist die Kapazität des Schnittes $\delta^{out}(U)$ definiert als*

$$c(\delta^{out}(U)) = \sum_{a \in \delta^{out}(U)} c(a).$$

Analog kann man $c(\delta^{in}(U))$ definieren. Es wird aber hier nicht benötigt.

- (4) *Das Problem des Finden eines minimalen s - t -Schnittes (Min-Cut-Problem) ist wie folgt definiert: Gegeben sei ein gerichteter Graph $D = (V, A)$, eine Quelle $s \in V$, eine Senke $t \in V$ und eine Kapazitätsfunktion $c \in \mathbb{R}_{\geq 0}^A$ und gesucht ist die Lösung des Minimierungsproblems*

$$\min\{c(\delta^{\text{out}}(U)) : U \subseteq V, s \in U, t \notin U\}.$$

Nun können wir das Max-Flow-Min-Cut Theorem von Ford und Fulkerson formulieren, eine weitere Min-Max-Charakterisierung. Dieser Satz gehört zu den wichtigsten Aussagen des Operations Research.

SATZ 1.3. (*Max-Flow-Min-Cut-Theorem; Ford-Fulkerson, 1954*)

Es seien ein gerichteter Graph $D = (V, A)$, zwei Knoten $s, t \in V$ und eine Kapazitätsfunktion $c \in \mathbb{R}_{\geq 0}^A$ gegeben. Dann gilt

$$\begin{aligned} & \max \left\{ \text{value}(f) : f \in \mathbb{R}_{\geq 0}^A \text{ ist } s\text{-}t\text{-Fluss, } f \leq c \right\} \\ & = \min\{c(\delta^{\text{out}}(U)) : U \subseteq V, s \in U, t \notin U\}. \end{aligned}$$

Wichtiger Zusatz: Falls c ganzzahlig ist, das heißt $c(a) \in \mathbb{Z}$ für alle $a \in A$, dann gibt es einen ganzzahligen maximalen s - t -Fluss $f \in \mathbb{Z}_{\geq 0}^A$.

Der Zusatz ist sowohl von theoretischer als auch von praktischer Bedeutung. Zum Beispiel kann man mit Hilfe des Zusatzes einsehen, dass der Satz von König ein Spezialfall des Max-Flow-Min-Cut-Theorems ist. In der Praxis hat man es oft mit zu transportierenden Gütern zu tun, die man nicht teilen kann, so dass man ausschließlich mit ganzzahligen Flüssen arbeiten möchte.

LEMMA 1.4. *Es seien ein gerichteter Graph $D = (V, A)$, zwei Knoten $s, t \in V$ und eine Kapazitätsfunktion $c \in \mathbb{R}_{\geq 0}^A$ gegeben. Sei $f \in \mathbb{R}_{\geq 0}^A$ ein s - t -Fluss, der durch c beschränkt ist. Sei $U \subseteq V$ mit $s \in U, t \notin U$, so dass $\delta^{\text{out}}(U)$ ein s - t -Schnitt ist. Dann gilt*

$$(1) \quad \text{value}(f) \leq c(\delta^{\text{out}}(U)).$$

Es gilt Gleichheit in (1) genau dann, wenn

$$\begin{aligned} f(a) &= c(a) \quad \text{für alle } a \in \delta^{\text{out}}(U), \\ f(a) &= 0 \quad \text{für alle } a \in \delta^{\text{in}}(U). \end{aligned}$$

DEFINITION 1.5. *Es sei $D = (V, A)$ ein gerichteter Graph und $a = (u, v)$ eine Kante. Definiere*

$$a^{-1} = (v, u) \quad \text{und} \quad A^{-1} = \{a^{-1} : a \in A\}.$$

Sei f ein s - t -Fluss und c eine Kapazitätsfunktion. Definiere den Residualgraph $D_f = (V, A_f)$ durch

$$A_f = \{a \in A : f(a) < c(a)\} \cup \{a^{-1} \in A^{-1} : f(a) > 0\}.$$

LEMMA 1.6. *Es sei f ein s - t -Fluss, der durch die Kapazitätsfunktion c beschränkt ist. Angenommen der Residualgraph D_f enthält keinen gerichteten Weg von s nach t . Es sei $U \subseteq V$ die Menge der Knoten, die in D_f von s aus erreichbar sind. Dann gilt*

$$\text{value}(f) = c(\delta^{\text{out}}(U)).$$

Insbesondere ist f nach Lemma 1.4 maximal.

DEFINITION 1.7. Angenommen es gibt in D_f einen Weg P von s nach t . Definiere den charakteristischen Vektor $\chi^P \in \mathbb{R}^A$ von P durch

$$\chi^P(a) = \begin{cases} 1, & \text{falls } P \text{ die Kante } a \text{ durchläuft,} \\ -1, & \text{falls } P \text{ die Kante } a^{-1} \text{ durchläuft,} \\ 0, & \text{sonst.} \end{cases}$$

Input : Gerichteter Graph $D = (V, A)$, $s, t \in V$, $c \in \mathbb{R}_{\geq 0}^A$
Output : Maximaler s - t -Fluss $f \in \mathbb{R}_{\geq 0}^A$ mit $f \leq c$
 Setze $f = 0$
while \exists gerichteter Weg P von s nach t in D_f **do**
 | Wähle $\varepsilon > 0$ maximal, so dass $0 \leq f + \varepsilon\chi^P \leq c$
 | $f = f + \varepsilon\chi^P$
end

Algorithmus 4 : Algorithmus von Ford und Fulkerson zur Bestimmung eines s - t -Flusses mit maximalem Wert

SATZ 1.8. Falls $c(a) \in \mathbb{Q}$ für allen Kanten $a \in A$, dann terminiert der Algorithmus von Ford und Fulkerson in endlich vielen Schritten; sonst im Allgemeinen nicht.

Es gibt tatsächlich Beispiele mit $c(a) \in \mathbb{R} \setminus \mathbb{Q}$, so dass der Algorithmus nicht terminiert, siehe z.B. das Buch über Combinatorial Optimization von Schrijver, Abschnitt 10.4a.

Falls die Wahl des Weges P im Algorithmus nicht eindeutig ist, kann es durch unglückliche Wahl passieren, dass die Anzahl der benötigten Iteration sehr groß wird und z.B. auch von den Werten der Kapazitätsfunktion abhängt.

Dinits (1970) und Edmonds-Karp (1972) haben bewiesen, dass das nicht passieren kann, wenn die gewählten s - t -Wege immer minimale Kantenzahl haben. Sie haben gezeigt, dass falls in jeder Iteration des Algorithmus von Ford und Fulkerson ein s - t -Weg in D_f mit minimaler Kantenzahl ausgewählt wird, beträgt die Anzahl der Iterationen höchstens $|V| \cdot |A|$. Insbesondere ist sie unabhängig von der Kapazitätsfunktion.

Teil B

**Einführung in die konvexe und
lineare Optimierung**

Konvexität

Viele Optimierungsprobleme des Operations Research lassen sich als konvexe bzw. als lineare Programme formulieren. Ein lineares Programm ist ein Optimierungsproblem von der Form

$$\max\{c^T x : x \in \mathbb{R}^n, Ax \leq b\},$$

wobei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ gegeben sind und wobei $Ax \leq b$ eine Kurzschreibweise für die m linearen Ungleichungen $(Ax)_j \leq b_j$ mit $j = 1, \dots, m$ ist.

Ziel wird sein, die Menge der zulässigen Lösungen $\{x \in \mathbb{R}^n : Ax \leq b\}$ geometrisch zu verstehen und zu einer geometrischen Erklärung von allgemeinen Min-Max-Charakterisierungen zu gelangen.

Um diese Ziel zu erreichen, werden wir die lineare Algebra über dem Körper der reellen Zahlen durch Nichtnegativitätsbedingungen erweitern.

1. Grundbegriffe aus Topologie und Geometrie

Generalvoraussetzung: Sei E ein n -dimensionaler euklidischer \mathbb{R} -Vektorraum mit Skalarprodukt $\langle x, y \rangle$ und Norm $\|x\| = \sqrt{\langle x, x \rangle}$. Wir wissen aus der linearen Algebra, dass wir durch Auswahl einer Orthonormalbasis von E annehmen können, dass $E = \mathbb{R}^n$ und $\langle x, y \rangle = x^T y$ ist.

DEFINITION 1.1. (1) Die (abgeschlossene) Kugel $B(x, r)$ mit Mittelpunkt $x \in \mathbb{R}^n$ und Radius $r \geq 0$ ist definiert als

$$B(x, r) = \{y \in \mathbb{R}^n : \|x - y\| \leq r\}.$$

(2) Sei $A \subseteq \mathbb{R}^n$ eine Menge. Ein Punkt $x \in A$ heißt innerer Punkt von A , falls es ein $\varepsilon > 0$ gibt mit $B(x, \varepsilon) \subseteq A$. Das Innere von A ist

$$\text{int } A = \{x \in A : x \text{ innerer Punkt von } A\}.$$

Die Menge A heißt offen, falls $A = \text{int } A$ ist.

(3) Die Menge A heißt abgeschlossen, falls $\mathbb{R}^n \setminus A$ offen ist. Der Abschluss von A ist

$$\bar{A} = \bigcap_{B \supseteq A, B \text{ abgeschlossen}} B.$$

(4) Die Menge A heißt kompakt, wenn jede Folge $(x_i)_{i \in \mathbb{N}}$ bestehend aus Elementen aus A eine konvergente Teilfolge besitzt, deren Grenzwert in A liegt.

(5) Die Menge A heißt beschränkt, falls es ein $r \geq 0$ gibt, so dass $A \subseteq B(0, r)$.

Aus der Analysis ist bekannt:

- (1) Die Menge A ist abgeschlossen genau dann, wenn für jede konvergente Folge $(x_i)_{i \in \mathbb{N}}$ mit $x_i \in A$ gilt $\lim_{i \rightarrow \infty} x_i \in A$.
- (2) Die Menge A ist kompakt genau dann, wenn sie abgeschlossen und beschränkt ist.

DEFINITION 1.2. Sei $A \subseteq \mathbb{R}^n$. Der Rand von A ist definiert als

$$\partial A = \{x \in \mathbb{R}^n : \forall \varepsilon > 0 : B(x, \varepsilon) \cap A \neq \emptyset \text{ und } B(x, \varepsilon) \cap (\mathbb{R}^n \setminus A) \neq \emptyset\}.$$

Man kann zeigen, dass ∂A abgeschlossen ist, und dass die Beziehungen $\overline{A} = A \cup \partial A$ und $\partial A = \overline{A} \setminus (\text{int } A)$ gelten.

DEFINITION 1.3. (1) Der Punkt $y \in \mathbb{R}^n$ heißt eine affine Kombination der Punkte $x_1, \dots, x_N \in \mathbb{R}^n$, falls es $\alpha_1, \dots, \alpha_N \in \mathbb{R}$ gibt, so dass

$$1 = \sum_{i=1}^N \alpha_i \quad \text{und} \quad y = \sum_{i=1}^N \alpha_i x_i$$

gilt.

(2) Die Punkte x_1, \dots, x_N heißen affin unabhängig, falls für alle $\alpha_1, \dots, \alpha_N \in \mathbb{R}$ mit

$$0 = \sum_{i=1}^N \alpha_i \quad \text{und} \quad 0 = \sum_{i=1}^N \alpha_i x_i$$

folgt, dass

$$0 = \alpha_1 = \alpha_2 = \dots = \alpha_N$$

gilt. Die Punkte x_1, \dots, x_N heißen affin abhängig, falls sie nicht affin unabhängig sind.

(3) Eine Menge $A \subseteq \mathbb{R}^n$ heißt affiner Unterraum, falls jede affine Kombination von Punkten aus A wieder ein Punkt in A ist. Aus der linearen Algebra ist bekannt, dass affine Unterräume, die nicht gleich der leeren Menge sind, immer von der Form $A = x + U$ sind, wobei $x \in \mathbb{R}^n$ ein Vektor und $U \subseteq \mathbb{R}^n$ ein Untervektorraum von \mathbb{R}^n ist. Die Dimension von einem affinen Unterraum $A = x + U$ ist definiert als $\dim A = \dim U$. Falls $A = \emptyset$ setzen wir $\dim A = -1$.

(4) Die affine Hülle von einer Menge $A \subseteq \mathbb{R}^n$ ist definiert als

$$\text{aff } A = \bigcap_{B \supseteq A, B \text{ affiner Unterraum}} B$$

Die Dimension von A ist definiert als $\dim A = \dim \text{aff } A$.

Man kann leicht beweisen, dass

$$\text{aff } A = \left\{ \sum_{i=1}^N \alpha_i x_i : N \in \mathbb{N}, x_1, \dots, x_N \in A, \alpha_1, \dots, \alpha_N \in \mathbb{R}, \sum_{i=1}^N \alpha_i = 1 \right\}$$

gilt. Wir werden einen sehr ähnlichen Beweis für die Definition der konvexen Hülle führen.

Ein affiner Unterraum der Dimension 0 ist ein Punkt. Affine Unterräume der Dimension 1 sind Geraden. Affine Unterräume der Dimension $n - 1$ werden Hyperebenen genannt. Man kann eine Hyperebene schreiben als

$$H_{c,\delta} = \{x \in \mathbb{R}^n : c^\top x = \delta\},$$

wobei $c \in \mathbb{R}^n \setminus \{0\}$ und $\delta \in \mathbb{R}$. Angenommen, der Vektor c ist normiert, d.h. es gelte $\|c\| = 1$. Dann gibt die Gleichung $c^\top x = \delta$ die Hessesche Normalenform der Geradengleichung im Fall $n = 2$ und die Hessesche Normalenform der Ebenengleichung im Fall $n = 3$ an. Dabei ist $|\delta|$ der Abstand der Hyperebene zum Ursprung.

2. Konvexe Mengen und konvexe Funktionen

DEFINITION 2.1. (1) Der Punkt $y \in \mathbb{R}^n$ heißt eine Konvexkombination der Punkte $x_1, \dots, x_N \in \mathbb{R}^n$, falls es $\alpha_1, \dots, \alpha_N \geq 0$ gibt, so dass

$$1 = \sum_{i=1}^N \alpha_i \quad \text{und} \quad y = \sum_{i=1}^N \alpha_i x_i$$

gilt.

- (2) Eine Menge $C \subseteq \mathbb{R}^n$ heißt konvex, wenn jede Konvexkombination von Punkten aus C wieder ein Punkt aus C ergibt, d.h.

$$\forall N \in \mathbb{N} \forall x_1, \dots, x_N \in C \forall \alpha_1, \dots, \alpha_N \geq 0 \text{ mit } \sum_{i=1}^N \alpha_i = 1 \implies \sum_{i=1}^N \alpha_i x_i \in C.$$

- (3) Sei $A \subseteq \mathbb{R}^n$. Die konvexe Hülle von A ist definiert als

$$\text{conv } A = \bigcap_{B \supseteq A, B \text{ konvex}} B.$$

- (4) Die Verbindungsstrecke zwischen zwei Punkten $x, y \in \mathbb{R}^n$ ist definiert als

$$[x, y] = \text{conv}\{x, y\}.$$

SATZ 2.2. (1) Sei $A \subseteq \mathbb{R}^n$. Dann ist

$$\text{conv } A = \left\{ \sum_{i=1}^N \alpha_i x_i : N \in \mathbb{N}, x_1, \dots, x_N \in A, \alpha_1, \dots, \alpha_N \geq 0, \sum_{i=1}^N \alpha_i = 1 \right\}.$$

Insbesondere gilt $[x, y] = \{(1 - \alpha)x + \alpha y : \alpha \in [0, 1]\}$.

- (2) Eine Menge $C \subseteq \mathbb{R}^n$ ist genau dann konvex, wenn für alle $x, y \in C$ gilt $[x, y] \subseteq C$.

SATZ 2.3. (Carathéodory)

Sei $A \subseteq \mathbb{R}^n$ und $y \in \text{conv } A$. Dann gibt es affin unabhängige Punkte $x_1, \dots, x_N \in A$ mit $y \in \text{conv}\{x_1, \dots, x_N\}$. Insbesondere gilt $N \leq \dim A + 1 \leq n + 1$.

DEFINITION 2.4. Sei $C \subseteq \mathbb{R}^n$ eine konvexe Menge. Eine Funktion $f: C \rightarrow \mathbb{R}$ heißt konvex, wenn ihr Epigraph, der durch

$$\text{epi } f = \{(x, \beta) \in C \times \mathbb{R} : f(x) \leq \beta\} \subseteq \mathbb{R}^{n+1}$$

definiert ist, eine konvexe Menge ist. Die Funktion f heißt konkav, wenn $-f$ konvex ist.

SATZ 2.5. (Ungleichung von Jensen)

Eine Funktion $f: C \rightarrow \mathbb{R}$ ist genau dann konvex, wenn für alle $x, y \in C$ und alle $\alpha \in [0, 1]$ stets die Ungleichung

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

gilt.

Ein sehr nützliches Konvexitätskriterium aus der Analysis:

SATZ 2.6. Es sei $C \subseteq \mathbb{R}^n$ eine offene und konvexe Menge. Sei $f: C \rightarrow \mathbb{R}$ eine zweifach stetig differenzierbare Funktion. Dann ist die Funktion f genau dann konvex, wenn die Hessematrix

$$H(f)(a) = \left(\frac{\partial^2}{\partial x_i \partial x_j} f(a) \right)_{i,j=1,\dots,n} \in \mathbb{R}^{n \times n}$$

für jedes $a \in C$ eine positiv semidefinite Matrix ist.

DEFINITION 2.7. Es seien $C \subseteq \mathbb{R}^n$ eine konvexe Menge und $f: C \rightarrow \mathbb{R}$ eine konvexe Funktion. Diese definieren ein konvexes Optimierungsproblem

$$\inf\{f(x) : x \in C\}.$$

Die Menge C heißt die Menge der zulässigen/gültigen Lösungen, die Funktion f heißt die Zielfunktion des konvexen Optimierungsproblems.

SATZ 2.8. *Wie oben seien $C \subseteq \mathbb{R}^n$ eine konvexe Menge und $f: C \rightarrow \mathbb{R}$ eine konvexe Funktion. Angenommen $x_0 \in C$ sei ein lokales Minimum des konvexen Optimierungsproblems $\inf\{f(x) : x \in C\}$, d.h. es gibt ein $\varepsilon > 0$, so dass*

$$f(x_0) = \inf\{f(x) : x \in C \cap B(x_0, \varepsilon)\}$$

gilt. Dann ist x_0 auch ein globales Minimum, d.h. es gilt

$$f(x_0) = \inf\{f(x) : x \in C\}.$$

3. Trennungssätze

In diesem Abschnitt werden wir beweisen, dass konvexe Mengen eine fundamentale Eigenschaft besitzen: Punkte, die außerhalb der konvexen Menge liegen können mit Hilfe einer Hyperebene von der konvexen Mengen getrennt werden.

DEFINITION 3.1. (1) *Eine Menge $H \subseteq \mathbb{R}^n$ heißt (affine) Hyperebene, falls es einen Vektor $c \in \mathbb{R}^n \setminus \{0\}$ und ein $\delta \in \mathbb{R}$ gibt mit*

$$H = \{x \in \mathbb{R}^n : c^\top x = \delta\}.$$

(2) *Die abgeschlossenen und konvexen Mengen*

$$H^+ = \{x \in \mathbb{R}^n : c^\top x \geq \delta\}$$

$$H^- = \{x \in \mathbb{R}^n : c^\top x \leq \delta\}$$

heißen Halbräume.

DEFINITION 3.2. (1) *Seien $C, D \subseteq \mathbb{R}^n$. Eine Hyperebene H heißt Trennhyperebene von C und D , falls $C \subseteq H^-$ und $D \subseteq H^+$ (oder umgekehrt).*

(2) *Sei $C \subseteq \mathbb{R}^n$. Eine Hyperebene H heißt Stützhyperebene von C , falls $C \subseteq H^-$ und $C \cap H \neq \emptyset$.*

LEMMA 3.3. *Sei $C \subseteq \mathbb{R}^n$ eine abgeschlossene und konvexe Menge, die nicht leer ist. Sei $z \in \mathbb{R}^n$. Dann gibt es genau einen Punkt $y \in C$ mit der Eigenschaft*

$$\|y - z\| = \inf\{\|x - z\| : x \in C\}.$$

Dieser Punkt wird auch als metrische Projektion von z auf C bezeichnet; Notation: $y = \pi_C(z)$. Darüber hinaus gilt für alle $x \in C$ die Ungleichung

$$(z - y)^\top (x - y) \leq 0.$$

SATZ 3.4. *Sei $C \subseteq \mathbb{R}^n$ eine abgeschlossene und konvexe Menge, die nicht leer ist. Sei $z \notin C$. Dann gibt es eine Trennhyperebene H von $\{z\}$ und C .*

KOROLLAR 3.5. *Die im Beweis des vorhergehenden Satzes konstruierte Trennhyperebene*

$$H = \{x \in \mathbb{R}^n : c^\top x = \delta\} \quad \text{mit} \quad c = z - y, \delta = c^\top y, y = \pi_C(z)$$

ist gleichzeitig eine Stützhyperebene von C . Man bekommt eine strikte Trennhyperebene, wenn man $\delta \in (c^\top z, c^\top y)$ wählt.

KOROLLAR 3.6. *Sei $C \subseteq \mathbb{R}^n$ eine abgeschlossene und konvexe Menge, die nicht leer ist. Die metrische Projektion π_C ist Lipschitzstetig mit Lipschitzkonstante 1. D.h. π_C ist eine Kontraktion.*

SATZ 3.7. *Sei $C \subseteq \mathbb{R}^n$ eine abgeschlossene und konvexe Menge, die nicht leer ist. Dann gibt es für jeden Randpunkt $y \in \partial C$ von C einen Punkt $z \notin C$ mit $\pi_C(z) = y$.*

KOROLLAR 3.8. *Sei $C \subseteq \mathbb{R}^n$ eine abgeschlossene und konvexe Menge, die nicht leer ist. Sei $y \in \partial C$. Dann gibt es eine Stützhyperebene H von C mit $y \in H$.*

SATZ 3.9. Seien $C, D \subseteq \mathbb{R}^n$ nichtleere, abgeschlossene und konvexe Mengen mit $C \cap D = \emptyset$. Dann gibt es ein $c \in \mathbb{R}^n \setminus \{0\}$ mit

$$\sup_{x \in C} c^\top x \leq \inf_{x \in D} c^\top x.$$

4. Repräsentation konvexer Mengen

In der informatiknahen Mathematik ist es immens wichtig, wie mathematische Objekte repräsentiert / dargestellt werden.

Äußere/Implizite Darstellung. Z.B. nützlich, um zu überprüfen, ob ein Punkt in einer konvexen Menge liegt.

SATZ 4.1. Sei $C \subseteq \mathbb{R}^n$ eine abgeschlossene und konvexe Menge, die nicht leer ist. Dann gilt

$$C = \bigcap_H H^-,$$

wobei H durch alle Stützhyperebenen von C läuft und wobei $C \subseteq H^-$ gilt.

Innere/Explizite Darstellung. Z.B. nützlich, um Punkte zu erzeugen, die in einer konvexen Menge liegen.

DEFINITION 4.2. Sei $C \subseteq \mathbb{R}^n$ eine Menge. Ein Punkt $z \in C$ heißt *Extrempunkt* von C , falls für alle $x, y \in C$ und alle $\alpha \in (0, 1)$ mit

$$x = (1 - \alpha)y + \alpha z$$

stets $x = y = z$ gilt. Die Menge aller Extrempunkte von C wird mit $\text{ext } C$ bezeichnet.

SATZ 4.3. (Minkowski; Krein-Milman)

Sei $C \subseteq \mathbb{R}^n$ eine kompakte und konvexe Menge. Dann gilt $C = \text{conv}(\text{ext}(C))$.

Polyedertheorie

1. Polyeder und Polytope

Besonders wichtige Repräsentationen sind die, die eine endliche Beschreibung zulassen.

DEFINITION 1.1. Eine Menge $P \subseteq \mathbb{R}^n$ heißt Polyeder, falls es eine Matrix $A \in \mathbb{R}^{m \times n}$ und einen Vektor $b \in \mathbb{R}^m$ gibt, so dass

$$P = \{x \in \mathbb{R}^n : Ax \leq b\}$$

gilt.

Polyeder sind also Durchschnitte endlich vieler Halbräume bzw. die Lösungsmengen von linearen Ungleichungssystemen.

DEFINITION 1.2. Eine Menge $P \subseteq \mathbb{R}^n$ heißt Polytop, falls es eine Teilmenge $X \subseteq \mathbb{R}^n$ mit $|X| < \infty$ gibt mit

$$P = \text{conv } X.$$

Sprechweise: Die Extrempunkte von Polyedern bzw. Polytopen heißen Ecken.

SATZ 1.3. Sei $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ein Polyeder. Sei $z \in P$. Dann ist z genau dann eine Ecke von P , wenn $\text{rang } A_z = n$ ist. Dabei ist A_z die Teilmatrix von A die aus den Zeilenvektoren a_j^\top von A besteht, für die Gleichheit $a_j^\top z = b_j$ gilt.

Die zur Definition von A_z verwendeten Gleichungen heißen auch die an z aktiven Ungleichungen des Systems $Az \leq b$.

KOROLLAR 1.4. Ein Polyeder hat nur endlich viele Ecken.

KOROLLAR 1.5. Sei P ein beschränktes Polyeder. Dann ist P die konvexe Hülle von endlich vielen Punkten. Insbesondere ist P ein Polytop.

Auch die Umkehrung gilt: Ein Polytop ist ein beschränktes Polyeder.

DEFINITION 1.6. Sei $A \subseteq \mathbb{R}^n$. Dann heißt

$$A^\times = \{y \in \mathbb{R}^n : x^\top y \leq 1 \text{ für alle } x \in A\}$$

die Polare von A .

LEMMA 1.7. Es seien $A, B \subseteq \mathbb{R}^n$.

- (1) Falls $A \subseteq B$, dann folgt $A^\times \supseteq B^\times$.
- (2) Für $\alpha > 0$ gilt

$$(\alpha A)^\times = \frac{1}{\alpha} A^\times,$$

wobei generell βB die Menge $\{\beta b : b \in B\}$ bezeichnet, wobei $\beta \in \mathbb{R}$.

- (3) $(B_n)^\times = B_n$, wobei $B_n = B(0, 1) = \{x \in \mathbb{R}^n : x^\top x \leq 1\}$ die n -dimensionale Einheitskugel ist.
- (4) Falls $P = \text{conv}\{x_1, \dots, x_t\}$ ist, dann folgt

$$P^\times = \{y \in \mathbb{R}^n : x_1^\top y \leq 1, \dots, x_t^\top y \leq 1\}.$$

SATZ 1.8. *Ein Polytop ist ein beschränktes Polyeder.*

THEOREM 1.9. (Minkowski, Weyl)

Eine Menge $P \subseteq \mathbb{R}^n$ ist genau dann ein Polytop, wenn P ein beschränktes Polyeder ist.

DEFINITION 1.10. *Seien $x_1, \dots, x_N \in \mathbb{R}^n$. Der durch x_1, \dots, x_N erzeugte Kegel $C \subseteq \mathbb{R}^n$ ist definiert als*

$$C = \text{cone}\{x_1, \dots, x_N\} = \left\{ \sum_{i=1}^N \alpha_i x_i : \alpha_1, \dots, \alpha_N \geq 0 \right\}.$$

BEISPIEL 1.11. *Der nichtnegative Orthant*

$$\mathbb{R}_{\geq 0}^n = \{x \in \mathbb{R}^n : x_1 \geq 0, \dots, x_n \geq 0\} = \text{cone}\{e_1, \dots, e_n\}$$

ist ein endlich erzeugter Kegel.

SATZ 1.12. *Eine Menge $C \subseteq \mathbb{R}^n$ ist ein endlich erzeugter Kegel genau dann, wenn es eine Matrix $A \in \mathbb{R}^{m \times n}$ gibt mit*

$$C = \{x \in \mathbb{R}^n : Ax \leq 0\}.$$

Insbesondere ist ein endlich erzeugter Kegel ein Polyeder und somit eine abgeschlossene und konvexe Menge.

SATZ 1.13. *Eine Menge $P \subseteq \mathbb{R}^n$ ist ein Polyeder, genau dann wenn, es Vektoren $x_1, \dots, x_s \in \mathbb{R}^n$ und $y_1, \dots, y_t \in \mathbb{R}^n$ gibt, so dass*

$$P = \text{conv}\{x_1, \dots, x_s\} + \text{cone}\{y_1, \dots, y_t\}$$

gilt. Dabei bezeichnet

$$A + B = \{a + b : a \in A, b \in B\}$$

die Minkowskisumme von zwei Mengen $A, B \subseteq \mathbb{R}^n$.

2. Das Lemma von Farkas

Das Lemma von Farkas ist ein Lösbarkeitskriterium für Systeme von linearen Ungleichungen.

SATZ 2.1. (Farkas Lemma)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ gegeben. Dann gibt es einen Vektor $x \in \mathbb{R}^n$ mit $x \geq 0$ und $Ax = b$ genau dann, wenn es keinen Vektor $y \in \mathbb{R}^m$ gibt, der $A^T y \geq 0$ und $b^T y < 0$ erfüllt.

KOROLLAR 2.2. (Variante von Farkas Lemma)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ gegeben. Dann gibt es einen Vektor $x \in \mathbb{R}^n$ mit $Ax \leq b$ genau dann, wenn es keinen Vektor $y \in \mathbb{R}^m$ gibt, der $y \geq 0$, $A^T y = 0$ und $b^T y < 0$ erfüllt.

3. Lineare Optimierung / Lineare Programmierung

DEFINITION 3.1. *Ein lineares Programm (LP) in primaler Standardform ist ein Maximierungsproblem von der Form*

$$\begin{aligned} (PLP) \quad & p^* = \sup c^T x \\ & x \in \mathbb{R}^n \\ & Ax \leq b, \end{aligned}$$

wobei $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ gegeben sind.

Das zugehörige lineare Programm in dualer Standardform ist ein Minimierungsproblem von der Form

$$\begin{aligned} d^* &= \inf b^\top y \\ \text{(DLP)} \quad & y \in \mathbb{R}^m \\ & y \geq 0 \\ & A^\top y = c. \end{aligned}$$

SATZ 3.2. Falls die Menge

$$P = \{x \in \mathbb{R}^n : Ax \leq b\}$$

der zulässigen Lösungen von (PLP) ein nicht-leeres beschränktes Polyeder ist, dann gibt es eine Ecke z von P , die eine optimale Lösung von (PLP) ist, d.h. es gilt

$$c^\top z \geq c^\top x \quad \text{für alle } x \in P.$$

SATZ 3.3. (Dualitätstheorie von linearen Programmen)

Betrachte das primale lineare Programm (PLP) und das dazu duale lineare Programm (DLP).

- (a) (schwache Dualität) Falls x zulässig für (PLP) und y zulässig für (DLP) ist, dann gilt die Ungleichung

$$c^\top x \leq b^\top y.$$

Insbesondere ist $p^* \leq d^*$.

- (b) (Komplementarität) Es seien x optimal für (PLP) und y optimal für (DLP) und es gelte $p^* = d^*$. Dann ist

$$(Ax - b)^\top y = 0.$$

Mit anderen Worten: falls $y_j \neq 0$ ist, dann ist $(Ax - b)_j = 0$ und falls $(Ax - b)_j \neq 0$, dann ist $y_j = 0$ für $j = 1, \dots, m$.

- (c) (Optimalitätsbedingung) Falls x zulässig für (PLP) und y zulässig für (DLP) und falls $p^* = d^*$ gilt, dann sind x, y optimal genau dann, wenn $(Ax - b)^\top y = 0$.
- (d) (starke Dualität) Falls (PLP) und (DLP) zulässige Lösungen besitzen, dann gilt $p^* = d^*$ und es gibt optimale Lösungen x und y .

KOROLLAR 3.4. Es gilt

$$\max\{c^\top x : x \geq 0, Ax = b\} = \min\{b^\top y : A^\top y - c \geq 0, y \in \mathbb{R}^m\}$$

falls die beide Mengen gültiger Lösungen nicht leer sind.

4. Das Eliminationsverfahren von Fourier und Motzkin

Bislang haben wir die Theorie der Systeme linearer Ungleichungen vor allem geometrisch betrachtet, ohne auf Algorithmen einzugehen. Auch haben wir mehrmals Parallelen zur linearen Algebra aufgezeigt. Der wohl wichtigste Algorithmus in der linearen Algebra ist das Gaußsche Eliminationsverfahren zur Lösung von Systemen linearer Gleichungen. Das Eliminationsverfahren von Fourier und Motzkin ist eine Variante des Gaußschen Eliminationsverfahren, mit dem man Systeme linearer Ungleichungen lösen kann.

Ein algorithmisches Grundproblem in der Polyedertheorie ist zu entscheiden, ob ein Polyeder $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ nicht leer ist. In der linearen Algebra hat man für ein ähnliches Problem, zu entscheiden, ob die Lösungsmenge $L = \{x \in \mathbb{R}^n : Ax = b\}$ eines linearen Gleichungssystems nicht leer ist, das Eliminationsverfahren von Gauß. Wir lernen hier ein ähnliches Verfahren für unser Problem kennen.

Gegeben seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Ziel ist es, $x \in \mathbb{R}^n$ mit $Ax \leq b$ zu finden bzw. zu entscheiden (mit mathematischer Sicherheit), dass es ein solches x nicht gibt.

Wir wollen dazu zunächst die Variable x_1 eliminieren. Finde $\tilde{A} \in \mathbb{R}^{\tilde{m} \times (n-1)}$, $\tilde{b} \in \mathbb{R}^{\tilde{m}}$, so dass

$$\exists x \in \mathbb{R}^n : Ax \leq b \Leftrightarrow \exists \tilde{x} \in \mathbb{R}^{n-1} : \tilde{A}\tilde{x} \leq \tilde{b}.$$

Dazu multiplizieren wird die Zeilen von A und die entsprechenden Einträge von b mit positiven Konstanten. Dann hat das System $Ax \leq b$ nach Ummummerierung der Zeilen folgende Gestalt:

$$(*) \quad \begin{bmatrix} 1 & (a'_1)^\top \\ \vdots & \vdots \\ 1 & (a'_r)^\top \\ -1 & (a'_{r+1})^\top \\ \vdots & \vdots \\ -1 & (a'_{r+s})^\top \\ 0 & (a'_{r+s+1})^\top \\ \vdots & \vdots \\ 0 & (a'_m)^\top \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix},$$

wobei $(a'_i)^\top \in \mathbb{R}^{1 \times (n-1)}$ die i -te Zeile von A ist, in der das erste Element gelöscht wurde (es kann passieren, dass $r = 0$ oder $s = 0$ ist). Betrachte die ersten r Bedingungen:

$$x_1 + (a'_i)^\top \underbrace{\begin{pmatrix} x_2 \\ \vdots \\ x_n \end{pmatrix}}_{=\tilde{x}} \leq b_i \Leftrightarrow x_1 \leq b_i - (a'_i)^\top \tilde{x}, \quad i = 1, \dots, r.$$

Genauso die nächsten s Bedingungen:

$$-x_1 + (a'_{r+j})^\top \tilde{x} \leq b_{r+j} \Leftrightarrow x_1 \geq (a'_{r+j})^\top \tilde{x} - b_{r+j}, \quad j = 1, \dots, s.$$

Zusammen gilt also

$$(**) \quad \sup_{j=1, \dots, s} (a'_{r+j})^\top \tilde{x} - b_{r+j} \leq x_1 \leq \inf_{i=1, \dots, r} b_i - (a'_i)^\top \tilde{x}.$$

(Falls $s = 0$, dann ist $\sup_{j=1, \dots, s} (a'_{r+j})^\top \tilde{x} - b_{r+j} = -\infty$ und falls $r = 0$, ist $\inf_{i=1, \dots, r} b_i - (a'_i)^\top \tilde{x} = +\infty$. In diesen Fällen ist also P in Richtung x_1 unbeschränkt.) Also kann man x_1 eliminieren und das System $(*)$ ist genau dann lösbar, wenn das System

$$\begin{aligned} (a'_{r+j})^\top \tilde{x} - b_{r+j} &\leq b_i - (a'_i)^\top \tilde{x}, \quad i = 1, \dots, r, \quad j = 1, \dots, s \\ (a'_i)^\top \tilde{x} &\leq b_i, \quad i = r + s + 1, \dots, m \end{aligned}$$

bzw. das System

$$(***) \quad \begin{aligned} ((a'_{r+j})^\top + (a'_i)^\top) \tilde{x} &\leq b_i + b_{r+j}, \quad i = 1, \dots, r, \quad j = 1, \dots, s \\ (a'_i)^\top \tilde{x} &\leq b_i, \quad i = r + s + 1, \dots, m. \end{aligned}$$

lösbar ist. Das neue System hat $r \cdot s + m - (r + s)$ viele Ungleichungen und $n - 1$ Variablen.

BEMERKUNG 4.1. (1) $(***)$ entspricht der Projektion des Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ entlang der x_1 -Achse.

(2) Eine Lösung \tilde{x} kann zu einer Lösung (x_1, \tilde{x}) von $(*)$ erweitert werden. Dazu muss x_1 die Ungleichungen $(**)$ erfüllen.

- (3) *Das Verfahren wird fortgesetzt, indem nun sukzessive die Variablen x_2, x_3, \dots eliminiert werden, bis man bei x_n angekommen ist.*
- (4) *Für x_n ist es offensichtlich, ob das finale System eine Lösung besitzt. Das finale System hat genau dann eine Lösung, wenn das Ursprungssystem (*) eine Lösung besitzt.*

Eine Anwendung: Wir wollen das LP

$$\begin{aligned} & \max c^\top x \\ \text{(LP)} \quad & x \in \mathbb{R}^n \\ & Ax \leq b \end{aligned}$$

mit dem Eliminationsverfahren von Fourier und Motzkin lösen. Dazu führen wir eine zusätzliche Variable λ ein und betrachten das System

$$Ax \leq b, \lambda \leq c^\top x.$$

Die Idee ist, dass λ dem größtmöglichen Wert der Zielfunktion $c^\top x$, also dem Maximum, so dass alle Ungleichungen erfüllt sind, entsprechen soll. Wegen $\lambda \leq c^\top x \iff \lambda - c^\top x \leq 0$, ist das System äquivalent zu

$$\begin{pmatrix} A & 0 \\ -c^\top & 1 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} \leq \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

Man kann nun das LP lösen, indem man eine Lösung $\begin{pmatrix} x \\ \lambda \end{pmatrix}$ von diesem System findet, so dass λ so groß wie möglich ist. Dazu eliminiert man x_1, \dots, x_n bis λ die letzte Variable ist. Dann wählt man λ so groß wie möglich.

Ganzzahlige lineare Optimierung und vollständig unimodulare Matrizen

1. Ganzzahlige lineare Programme

Viele Optimierungsprobleme des Operations Research lassen sich als *ganzzahlige lineare Programme* formulieren.

DEFINITION 1.1. Ein ganzzahliges lineares Programm (in Standardform) ist von der Form:

$$\begin{aligned} \max \quad & c^\top x \\ & x \in \mathbb{Z}^n \quad (\text{„}x \text{ ganzzahlig“}) \\ & Ax \leq b, \end{aligned}$$

wobei $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ gegeben sind. Ein ganzzahliges lineares Programm nennen wir auch *ILP* oder *IP* (= „integer linear program“).

Geometrisch bedeutet dies, dass wir eine lineare Funktion über die Menge $\mathbb{Z}^n \cap P$, wobei $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ein Polyeder ist, maximieren.

BEISPIEL 1.2. Es sei $G = (V, E)$ ein ungerichteter Graph mit Inzidenzmatrix $A \in \mathbb{R}^{V \times E}$. Dann ist

$$\begin{aligned} \nu(G) &= \max\{|M| : M \subseteq E \text{ Matching in } G\} \\ &= \max \left\{ \sum_{e \in E} x_e : x_e \geq 0, x_e \in \mathbb{Z} \forall e \in E, \sum_{e:v \in e} x_e \leq 1 \forall v \in V \right\} \\ &= \max \left\{ \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^\top x : x \geq 0, Ax \leq \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, x \in \mathbb{Z}^E \right\} \end{aligned}$$

DEFINITION 1.3. Ein Polyeder $P \subseteq \mathbb{R}^n$ heißt *ganzzahlig*, falls für alle $c \in \mathbb{R}^n$, für die $\sup\{c^\top x : x \in P\}$ endlich ist, das Maximum an einem ganzzahligen Vektor angenommen wird.

2. Vollständig unimodulare Matrizen

DEFINITION 2.1. Eine Matrix $A \in \mathbb{R}^{m \times n}$ heißt *vollständig-unimodular (VU)*, falls jeder ihrer Minoren (Determinanten quadratischer Teilmatrizen) gleich 0, -1 oder +1 ist.

Insbesondere gilt für die Einträge einer vollständig-unimodularen Matrix A : $A_{ij} \in \{0, -1, +1\}$.

SATZ 2.2. Sei $A \in \mathbb{R}^{m \times n}$ eine vollständig-unimodulare Matrix und sei $b \in \mathbb{Z}^m$. Dann ist jede Ecke z des Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ganzzahlig, d.h. es gilt $z \in \mathbb{Z}^n$.

SATZ 2.3. Sei $A \in \mathbb{R}^{m \times n}$ eine vollständig-unimodulare Matrix und sei $b \in \mathbb{Z}^m$. Dann ist $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ein ganzzahliges Polyeder.

KOROLLAR 2.4. Sei $A \in \mathbb{R}^{m \times n}$ eine vollständig-unimodulare Matrix und sei $b \in \mathbb{Z}^m$ und $c \in \mathbb{Z}^n$. Dann haben die beiden linearen Programme

$$\max\{c^T x : Ax \leq b\} = \min\{b^T y : y \geq 0, A^T y = c\}$$

ganzzahlige Lösungen, falls die Optima endlich sind.

3. Vollständig-unimodulare Matrizen und bipartite Graphen

SATZ 3.1. Sei $G = (V, E)$ ein ungerichteter Graph. Dann ist der Graph G bipartit genau dann, wenn seine Inzidenzmatrix $A \in \mathbb{R}^{V \times E}$ vollständig-unimodular ist.

KOROLLAR 3.2. (Matching-Theorem von König)
Sei $G = (V, E)$ ein bipartiter Graph. Dann gilt $\nu(G) = \tau(G)$.

DEFINITION 3.3. Sei $G = (V, E)$ ein ungerichteter Graph. Das Matchingpolytop von G ist die konvexe Hülle der charakteristischen Vektoren von Matchings in G :

$$M(G) = \text{conv}\{\chi^M : M \subseteq E \text{ Matching in } G\} \subseteq \mathbb{R}^E,$$

dabei ist für ein Matching $M \subseteq E$ von G der charakteristische Vektor χ_M definiert als

$$(\chi_M)_e = \begin{cases} 1, & \text{falls } e \in M, \\ 0, & \text{sonst.} \end{cases}$$

KOROLLAR 3.4. Sei $G = (V, E)$ ein bipartiter Graph. Dann gilt

$$M(G) = \{x \in \mathbb{R}^E : x \geq 0, Ax \leq 1\},$$

wobei A die Inzidenzmatrix von G ist.

BEISPIEL 3.5. Falls G ein Dreieck ist, dann ist gilt nur die Inklusion

$$M(G) \subseteq \{x \in \mathbb{R}^E : x \geq 0, Ax \leq 1\},$$

aber nicht die Gleichheit.

KOROLLAR 3.6. (Theorem von Egerváry, 1931)
Sei $G = (V, E)$ ein bipartiter Graph und $w \in \mathbb{Z}^E$ eine ganzzahlige Gewichtsfunktion. Dann ist

$$\begin{aligned} \nu_w(G) &= \max\{w(M) : M \subseteq E \text{ Matching in } G\} \\ &= \min \left\{ \sum_{v \in V} y_v : y \in \mathbb{Z}^V, y \geq 0, y_u + y_v \geq w(\{u, v\}) \forall \{u, v\} \in E \right\}. \end{aligned}$$

4. Vollständig-unimodulare Matrizen und gerichtete Graphen

DEFINITION 4.1. Die Inzidenzmatrix eines gerichteten Graphen $D = (V, A)$ ist die Matrix $M \in \mathbb{R}^{V \times A}$, die durch

$$M_{v,a} = \begin{cases} +1, & \text{falls } a \in \delta^{in}(v), \\ -1, & \text{falls } a \in \delta^{out}(v), \\ 0, & \text{sonst.} \end{cases}$$

definiert ist.

Jede Spalte von M enthält genau eine $+1$ und genau eine -1 .

SATZ 4.2. Die Inzidenzmatrix eines gerichteten Graphen ist vollständig unimodular.

KOROLLAR 4.3. (*Max-Flow-Min-Cut Theorem*)

Sei $D = (V, A)$ ein gerichteter Graph, $s, t \in V$ und $c \in \mathbb{R}_{\geq 0}^A$ eine Kapazitätsfunktion. Dann gilt

$$\max\{\text{value}(f) : f \in \mathbb{R}_{\geq 0}^A \text{ s-t-Fluss}, f \leq c\} = \min\{c(\delta^{\text{out}}(U)) : U \subseteq V, s \in U, t \notin U\}.$$

5. Charakterisierung von vollständig unimodularen Matrizen

SATZ 5.1. (*Hoffman, Kruskal, 1956*)

Es sei $A \in \mathbb{Z}^{m \times n}$ eine ganzzahlige Matrix. Dann ist A vollständig unimodular genau dann, wenn für alle $b \in \mathbb{Z}^m$ das Polyeder $P_b = \{x \in \mathbb{R}^n : x \geq 0, Ax \leq b\}$ ganzzahlig ist.

Teil C

Algorithmen der linearen
Optimierung

Das Simplexverfahren

Wir möchten das lineares Programm

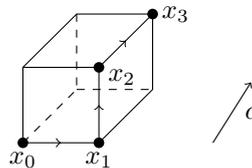
$$(LP) \quad \begin{aligned} p^* &= \max c^\top x \\ x &\in \mathbb{R}^n \\ Ax &\leq b. \end{aligned}$$

möglichst effizient algorithmisch lösen. Dabei wissen wir, dass, falls $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ein Polytop ist, dann wird das Maximum an einer Ecke von P angenommen wird.

Die geometrische Idee des Simplexverfahrens ist einfach: Finde eine Folge von Ecken $x_0, x_1, \dots, x_N \in P$, so dass

$$c^\top x_0 \leq c^\top x_1 \leq \dots \leq c^\top x_N = p^*$$

gilt.



Das Simplexverfahren wurde 1947 von dem US-amerikanischen Mathematiker George Dantzig entwickelt. Auch heutzutage ist es eines der wichtigsten und vielleicht sogar das meistgenutzte algorithmische Verfahren in der mathematischen Optimierung.

1. Simplexalgorithmus mit bekannter Startecke

Zunächst treffen wir die spezielle Annahme, dass das Polyeder P eine Ecke x_0 besitzt, die wir kennen. Später beseitigen wir diese spezielle Annahme.

Simplexalgorithmus

- Wähle ein Teilsystem $A_0 x \leq b_0$ von $Ax \leq b$ mit einer regulären quadratischen Matrix A_0 , wobei $A_0 x_0 = b_0$.
- Bestimme $u \in \mathbb{R}^m$ mit $c^\top = u^\top A$ und $u_i = 0$, falls Zeile i von A nicht zu A_0 gehört. Dazu berechne $c^\top A_0^{-1}$ und füge Nullen an den entsprechenden Stellen hinzu.

1. Fall: $u \geq 0$.

Dann ist x_0 optimal, weil u eine optimale duale Lösung ist. Denn:

$$\begin{aligned} c^\top x_0 &= u^\top A x_0 = u^\top b \geq \min\{y^\top b : y \geq 0, y^\top A = c^\top\} \\ &= \max\{c^\top x : Ax \leq b\}. \end{aligned}$$

2.Fall: $u \not\geq 0$.

Sei i der kleinste Index mit $u_i < 0$.

Wähle $y \in \mathbb{R}^n$ mit $a^\top y = 0$ für alle Zeilen a^\top von A_0 mit $a^\top \neq a_i^\top$

und $a_i^\top y = -1$.

(y ist die entsprechende Spalte von $-A_0^{-1}$).

2.a) $a^\top y \leq 0$ für alle Zeilen a^\top von A .

Dann ist $x_0 + \lambda y \in P \forall \lambda \geq 0$. Desweiteren ist

$$\begin{aligned} c^\top(x_0 + \lambda y) &= c^\top x_0 + \lambda c^\top y \\ &= c^\top x_0 + \lambda \underbrace{u^\top A y}_{=-u_i} \\ &= c^\top x_0 - \lambda u_i \rightarrow +\infty \text{ für } \lambda \rightarrow \infty. \end{aligned}$$

Das heißt das LP ist unbeschränkt.

2.b) $a^\top y > 0$ für eine Zeile a^\top von A .

Setze

$$\begin{aligned} \lambda_0 &= \max\{\lambda : x + \lambda y \in P\} \\ &= \min\left\{\frac{b_j - a_j^\top x_0}{a_j^\top y} : j = 1, \dots, m, a_j^\top y > 0\right\}, \end{aligned}$$

wobei die Gleichheit von Maximum und Minimum im Beweis des Theorems von Minkowski gezeigt wurde. Sei j der kleinste Index, an dem das Minimum angenommen wird. Definiere

$A_1 =$ Matrix, die man aus A_0 erhält,

indem man Zeile a_i^\top durch Zeile a_j^\top austauscht.

$$x_1 = x_0 + \lambda y.$$

Dann gilt $A_1 x_1 = b_1$.

- Gehe zum Anfang mit A_1, x_1 , anstelle von A_0, x_0 .
- Wiederhole diese Schritte, bis $u \geq 0$ oder bis klar ist, dass das LP unbeschränkt ist.

SATZ 1.1. *Der Simplexalgorithmus terminiert nach endlich vielen Schritten.*

BEWEIS. Wir bezeichnen die Variablen im k -ten Schritt mit $A_k, x_k, u_k, y_k, \lambda_{0,k}$. Es gilt

$$\begin{aligned} c^\top x_0 &\leq c^\top x_1 \leq \dots, \text{ und} \\ c^\top x_k &= c^\top x_{k+1} \Leftrightarrow x_k = x_{k+1}, \end{aligned}$$

weil

$$c^\top x_{k+1} = c^\top(x_k + \lambda_{0,k} y_k) \text{ mit } \lambda_{0,k} \geq 0$$

und

$$c^\top y_k = (-u_k)_i > 0.$$

Angenommen der Algorithmus landet in einer Endlosschleife. Dann gibt es k, l mit $k < l$ und $A_k = A_l$, weil es nur endlich viele verschiedene Teilmatrizen von A gibt. Dann

$$c^\top x_k = c^\top x_l, \text{ also } x_k = x_{k+1} = \dots = x_l.$$

Sei r der größte Index, so dass die Zeile a_r^\top in einer Iteration aus A_t genommen wird, wobei $t = k, k+1, \dots, l$. Dies passiere in Schritt p . Weil $A_k = A_l$, gibt es ein q , so dass a_r^\top wieder in A_q aufgenommen wird. Dann

$$k \leq p < q < l.$$

Dann gilt für $j > r$

$$a_j^\top \text{ kommt in } A_p \text{ vor} \iff a_j^\top \text{ kommt in } A_q \text{ vor.}$$

Es gilt

$$u_p^\top A y_q = c^\top y_q > 0.$$

Also gibt es ein j mit $(u_p)_j (a_j^\top y_q) > 0$. Aber:

1. Fall: a_j^\top gehört nicht zu A_p . Dann $(u_p)_j = 0$, Widerspruch.
2. Fall: a_j^\top gehört zu A_p .
 - a) $j > r$: Dann $a_j^\top y_q = 0$, Widerspruch.
 - b) $j = r$: Dann $(u_p)_j < 0$ und $a_j^\top y_q > 0$, Widerspruch.
 - c) $j < r$: Dann $(u_p)_j \geq 0$ und $a_j^\top y_q \leq 0$, Widerspruch.

□

2. Simplexalgorithmus ohne Startecke

Jetzt beschäftigen wir uns mit der Frage, wie man den Simplexalgorithmus startet, wenn man keine Ecke x_0 kennt.

OBdA: Das LP ist von der Form

$$\max\{c^\top x : x \geq 0, Ax \leq b\}.$$

Idee: Um eine Ecke von P zu finden, füge eine Extravariablen hinzu und stelle ein neues LP auf, das eine offensichtliche Ecke besitzt und dessen optimale Lösung eine Ecke von P liefert.

Extravariablen: $y \in \mathbb{R}^m, y \geq 0$.

Neues LP:

$$\min e^\top y$$

$$\begin{bmatrix} A & -I_m \\ -I_n & 0 \\ 0 & -I_m \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix},$$

mit $e = (1, \dots, 1)^\top$.

Offensichtliche Ecke:

$$x = 0, \quad y_j = \begin{cases} 0, & \text{falls } b_j \geq 0 \\ -b_j, & \text{falls } b_j < 0 \end{cases}, \quad j = 1, \dots, m.$$

Dann ist $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{n+m}$ eine Ecke von

$$P' = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} : Ax - y \leq b, x \geq 0, y \geq 0 \right\},$$

weil $\begin{pmatrix} x \\ y \end{pmatrix} \in P'$ und $\text{rang} \begin{bmatrix} A & -I_m \\ -I_n & 0 \\ 0 & -I_m \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = n + m$.

Jetzt kann man den Simplexalgorithmus mit der Startecke $\begin{pmatrix} x \\ y \end{pmatrix}$ verwenden, um das neue LP zu lösen. Sei $\begin{pmatrix} x^* \\ y^* \end{pmatrix}$ die Ecke von P' , die der Algorithmus liefert.

1.Fall: $e^T y^* > 0$.

Dann ist das Original-LP ungültig, denn

$$\nexists x \geq 0 : Ax \leq b, \text{ da } \exists j : y_j^* > 0 \text{ und } (Ax - y^*)_j \leq b_j.$$

2.Fall: $e^T y^* = 0$.

Dann $y^* = 0$ und x^* ist eine Ecke von P , weil $x^* \in P$ und $\text{rang} \begin{bmatrix} A \\ -I \end{bmatrix}_{x^*} = n$.

3. Zur praktischen und theoretischen Effizienz des Simplexalgorithmus

- + sehr schnell bei vielen praxisrelevanten Eingaben.
- + sehr gute Implementationen erhältlich (CPLEX, gurobi).
- Klee-Minty-Würfel (1972): Beispiel, dass der Algorithmus exponentiell viele Schritte im worst case benötigt.
- + Spielman-Teng (2004): „smoothed analysis“: Algorithmus ist polynomiell, falls die Eingabe leicht, zufällig „gestört“ ist.
 - offenes Problem („polynomielle Hirsch-Vermutung“): Ist der maximale Abstand zwischen zwei Ecken polynomiell in m, n ?

Die Ellipsoidmethode

1. Grundlagen zu Ellipsoiden

DEFINITION 1.1. Eine positiv definite Matrix $A \in \mathbb{R}^{n \times n}$ und ein Vektor $x \in \mathbb{R}^n$ definieren das Ellipsoid

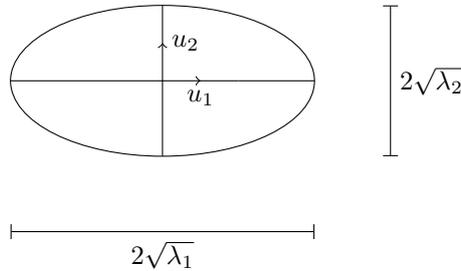
$$\mathcal{E}(A, x) = \{y \in \mathbb{R}^n : (y - x)^T A^{-1} (y - x) \leq 1\}.$$

Zum Beispiel ist $\mathcal{E}(r^2 I_n, 0) = rB_n$, wobei $B_n = \{y \in \mathbb{R}^n : y^T y \leq 1\}$ die n -dimensionale Einheitskugel ist.

Mit Hilfe der Hauptachsentransformation/Spektralzerlegung der positiv definiten Matrix $A \in \mathbb{R}^{n \times n}$ erhält man:

$$A = \sum_{i=1}^n \lambda_i u_i u_i^T,$$

wobei $\lambda_1, \dots, \lambda_n > 0$ die Eigenwerte von A sind, und $u_1, \dots, u_n \in \mathbb{R}^n$ eine Orthonormalbasis bilden, bestehend aus den zugehörigen Eigenvektoren. Diese Darstellung von A heißt *Spektralzerlegung*. Die Matrix A definiert ein Ellipsoid wie folgt: Die u_i bestimmen die Richtungen der Hauptachsen und die Länge der jeweiligen Hauptachse ist durch $2\sqrt{\lambda_i}$ gegeben.



SATZ 1.2. Das Volumen von $\mathcal{E}(A, x)$ ist

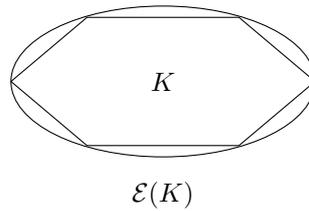
$$\text{vol } \mathcal{E}(A, x) = \sqrt{\det A} \cdot \text{vol } B_n,$$

wobei $\text{vol } B_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$, mit der Gammafunktion Γ , die für unsere Zwecke ausreichend definiert ist durch

$$\Gamma(1/2) = \sqrt{\pi}, \quad \Gamma(1) = 1, \quad \Gamma(x+1) = x \cdot \Gamma(x).$$

Der Beweis dieser Volumenformel kann mit Hilfe einer einfachen Anwendung des Transformationsatzes für n -dimensionale Integrale gegeben werden.

SATZ 1.3. Sei $K \subseteq \mathbb{R}^n$ eine konvexe und kompakte Menge, die nicht leer ist. Dann existiert ein eindeutiges Ellipsoid $\mathcal{E}(K)$ mit $K \subseteq \mathcal{E}(K)$ und mit minimalem Volumen (die beste äußere, ellipsoidförmige Approximation von K). Dieses Ellipsoid $\mathcal{E}(K)$ heißt das Loewner-John-Ellipsoid von K .

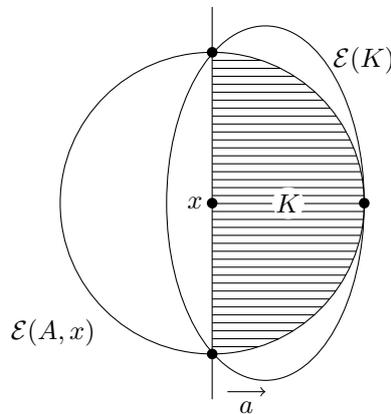


BEWEIS. → Vorlesung „Konvexe Optimierung“.

□

Hier: Spezialfall des Halbellipsoids

$$K = \mathcal{E}(A, x) \cap \{y \in \mathbb{R}^n : a^T y \geq a^T x\} \text{ mit } a \neq 0.$$



LEMMA 1.4. Sei $A \in \mathbb{R}^{n \times n}$ positiv definit, $x, a \in \mathbb{R}^n, a \neq 0$. Dann

$$\mathcal{E}(\mathcal{E}(A, x) \cap \{y \in \mathbb{R}^n : a^T y \geq a^T x\}) = \mathcal{E}(A', x')$$

mit

$$A' = \frac{n^2}{n^2 - 1} \left(A - \frac{2}{n+1} bb^T \right), \quad x' = x + \frac{1}{n} b, \quad b = \frac{1}{\sqrt{a^T A a}} A a.$$

Desweiteren gilt

$$\frac{\text{vol } \mathcal{E}(A', x')}{\text{vol } \mathcal{E}(A, x)} < e^{-\frac{1}{2(n+1)}} < 1.$$

BEWEIS. Anleitung:

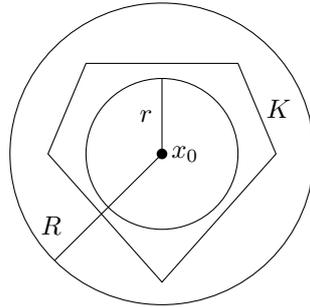
- $\mathcal{E}(A, x) \cap \{y \in \mathbb{R}^n : a^T y \geq a^T x\} \subseteq \mathcal{E}(A', x')$.
- Dass $\mathcal{E}(A', x')$ tatsächlich das Loewner-John-Ellipsoid ist, erhält man durch elementares Nachrechnen (etwas fummelig).
- Die Aussage über den Quotient der Volumina erhält man ebenfalls durch elementares Nachrechnen (etwas leichter).

□

2. Trennen und Optimieren

Hier werden wir folgende Generalvoraussetzung verwenden: Es sei \mathcal{K} eine Klasse von konvexen und kompakten Mengen. Angenommen für jedes $K \in \mathcal{K}$ kennen wir $x_0 \in \mathbb{R}^n, r, R > 0$ mit

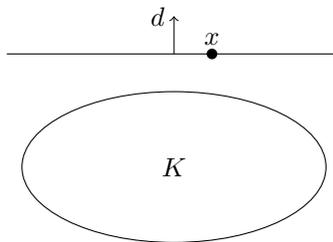
$$x_0 + rB_n \subseteq K \subseteq x_0 + RB_n.$$



DEFINITION 2.1. *Trennungsproblem:*

Eingabe: $x \in \mathbb{R}^n$, $K \in \mathcal{K}$.

Ausgabe: „ $x \in K$ “ oder $d \in \mathbb{R}^n$ mit $d^\top x > \max_{y \in K} d^\top y$.



Die Hyperebene $\{y \in \mathbb{R}^n : d^\top y = d^\top x\}$ trennt x und K .

Falls $K = \{x \in \mathbb{R}^n : Ax \leq b\}$ ein (beschränktes) Polyeder ist, dann kann man das Trennungsproblem wie folgt lösen: Überprüfe für gegebenes x alle linearen Ungleichungen

$$a_i^\top x \leq b_i \quad i = 1, \dots, m.$$

Dann ist $x \in K$ genau dann, wenn alle Ungleichungen erfüllt sind. Falls $a_i^\top x > b_i$, wähle $d = a_i$.

DEFINITION 2.2. *Optimierungsproblem:*

Eingabe: $c \in \mathbb{R}^n$, $\|c\| = 1$, $\varepsilon > 0$, $K \in \mathcal{K}$.

Ausgabe: $x \in K$ mit $c^\top x \geq \max_{y \in K} c^\top y - \varepsilon$.

THEOREM 2.3. (Grötschel, Lovász, Schrijver, 1981; basierend auf Ellipsoidmethode)

Für eine Klasse \mathcal{K} lässt sich das Optimierungsproblem durch N -faches Lösen des Trennungsproblems lösen, wobei N so gewählt ist, dass

$$2 \frac{R^2}{r} e^{-\frac{N}{2(n+1)^n}} \leq \varepsilon$$

gilt.

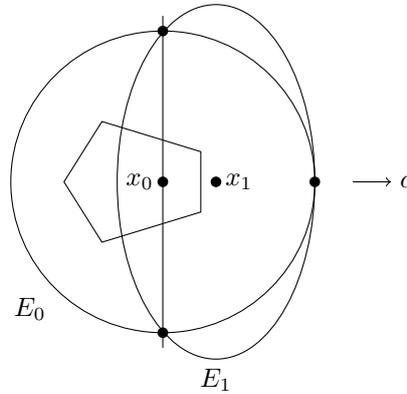
```

 $E_0 = \mathcal{E}(R^2 I_n, x_0)$ 
for  $k = 0$  to  $N - 1$  do
  Löse Trennungsproblem für  $x_k$ , den Mittelpunkt von  $E_k$ 
  if  $x_k \in K$  then
    |  $a = c$ 
  end
  else
    |  $a = -d$ 
  end
   $E_{k+1} = \mathcal{E}(E_k \cap \{y \in \mathbb{R}^n : a^\top y \geq a^\top x_k\})$ .
end

```

Algorithmus 5 : Ellipsoidmethode

3. Ellipsoidmethode



LEMMA 3.1. *Seien*

$$\xi_k = \max\{c^\top x_j : 0 \leq j \leq k, x_j \in K\}$$

und

$$K_k = K \cap \{x \in \mathbb{R}^n : c^\top x \geq \xi_k\}.$$

Dann gilt $K_k \subseteq E_k$.

BEWEIS. Der Beweis kann per Induktion nach k geführt werden und ist eine leichte Übungsaufgabe. \square

LEMMA 3.2. *Sei $k \in \mathbb{N}$ und sei j ein Index mit $0 \leq j < k$ und $\xi_k = c^\top x_j$. Dann gilt*

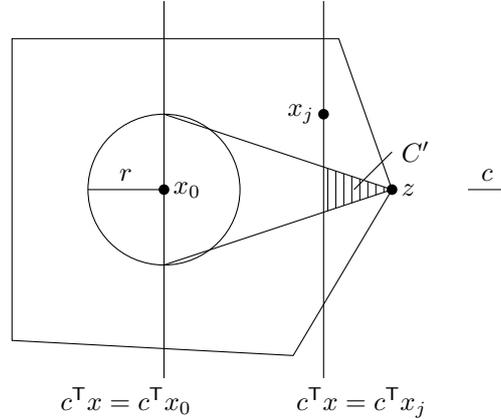
$$c^\top x_j \geq \max_{y \in K} c^\top y - 2 \frac{R^2}{r} e^{-\frac{k}{2(n+1)n}}.$$

BEWEIS. Sei $z \in K$ eine optimale Lösung, also $c^\top z = \max_{y \in K} c^\top y$. Betrachte den runden Kegelstumpf C mit Spitze z und Basis

$$x_0 + rB_n \cap \{y \in \mathbb{R}^n : c^\top y = c^\top x_0\}.$$

Sei

$$C' = C \cap \{x \in \mathbb{R}^n : c^\top x \geq c^\top x_j\}.$$



Das Volumen von C' ist

$$\text{vol } C' = \frac{r^{n-1} \cdot \text{vol } B_{n-1}}{n} (c^T z - c^T x_0) \left(\frac{c^T z - c^T x_j}{c^T z - c^T x_0} \right)^n$$

Nach dem vorhergehenden Lemma ist $C' \subseteq K_k \subseteq E_k$. Also

$$\begin{aligned} \text{vol } C' &\leq \text{vol } E_k \\ &\leq \text{vol } E_0 \cdot \left(e^{-\frac{1}{2n+1}} \right)^k \\ &< R^n \text{vol } B_n \cdot e^{-\frac{k}{2n+1}}. \end{aligned}$$

Die Ungleichung von Cauchy-Schwarz ergibt:

$$|c^T z - c^T x_0| \leq \underbrace{\|c\|}_{=1} \cdot \underbrace{\|z - x_0\|}_{\leq R} \leq R.$$

Zusammen:

$$(c^T z - c^T x_j)^n \leq R^n \text{vol } B_n e^{-\frac{k}{2n+1}} \cdot \frac{n \overbrace{(c^T z - c^T x_0)^{n-1}}^{\leq R}}{r^{n-1} \text{vol } B_{n-1}}.$$

Also, da $1 \leq \frac{R}{r}$:

$$c^T z - c^T x_j \leq \left(\frac{n \text{vol } B_n}{\text{vol } B_{n-1}} \right)^{\frac{1}{n}} \frac{R^2}{r} e^{-\frac{k}{2(n+1)n}}.$$

□

Aus dem letzten Lemma folgt das Theorem von Grötschel, Lovász und Schrijver unmittelbar.

Der hier vorgestellte Algorithmus ist an einer Stelle sehr idealisiert: Wir haben angenommen, dass wir mit unendlicher Genauigkeit rechnen können; aber in der Definition von b , das zur Bestimmung von E_k verwendet wird, wird eine Wurzel gezogen. Der Algorithmus kann leicht verändert werden, so dass er mit vorgegebener Bitkomplexität läuft. Die Analyse wird aber deutlich technischer.