Rumpfskript zur Vorlesung

Einführung in die Mathematik des Operations Research

Sommersemester 2017

Prof. Dr. Frank Vallentin

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

Inhaltsverzeichnis

Teil A

Graphen und Netzwerke, kombinatorische Algorithmen

KAPITEL 1

Minimale Spannbäume

1. Ungerichtete Graphen

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). 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 $v \in e$.*

DEFINITION 1.2. *Sei $G = (V, E)$ ein Graph. Eine Kantenfolge (path) P in G ist ein Tupel der Form*

$$P = (v_0, e_1, v_1, e_2, v_2, e_3, \ldots, e_m, v_m), \quad m \in \{0, 1, 2, \ldots\},$$

wobei $v_0, \ldots, v_m \in V$ und $e_i = \{v_{i-1}, v_i\} \in E$ für $i = 1, \ldots, m$. Der Knoten v_0 heißt Startknoten und der Knoten v_m heißt Endknoten der Kantenfolge P. Eine Kantenfolge P heißt $v_0 - v_m$-Weg (oder kurz: Weg), falls $|\{v_0, v_1, \ldots, v_m\}| = m+1$. Eine Kantenfolge P heißt Kreis, falls $v_0 = v_m$ und $|\{v_0, v_1, \ldots, v_m\}| = m$.

DEFINITION 1.3. *Sei $G = (V, E)$ ein Graph. Zwei Knoten $v, w \in V$ heißen wegzusammenhängend, falls es in G einen $v-w$-Weg gibt. Dies definiert eine Äquivalenzrelation auf V und ihre Äquivalenzklassen heißen Zusammenhangskomponenten. Falls G nur eine Zusammenhangskomponente besitzt, so heißt G zusammenhängend.*

DEFINITION 1.4. *Ein Graph $G = (V, E)$ heißt Wald, falls es in G keine Kreise gibt. Ein Baum (tree) ist ein zusammenhängender Wald. Ein Graph $H = (V, T)$ heißt Spannbaum von G, falls $T \subseteq E$ und falls H ein Baum ist.*

DEFINITION 1.5. *Sei $G = (V, E)$ ein Graph und sei $l \colon E \to \mathbb{R}$ eine Kantenlängenfunktion (Notation: $l \in \mathbb{R}^E$). Ein Graph $H = (V, T)$ heißt ein minimaler Spannbaum von G und l, wenn für alle Spannbäume $H' = (V, T')$ von G gilt:*

$$\sum_{e \in T} l(e) \leq \sum_{e \in T'} l(e).$$

SATZ 1.6. *Sei $G = (V, E)$ ein Graph. Die folgenden Aussagen sind äquivalent:*

(a) *G ist ein Baum.*
(b) *Für alle $v, w \in V$ gibt es einen eindeutigen $v - w$-Weg.*

DEFINITION 1.7. *Sei $G = (V, E)$ ein Graph.*

(a) *Die Inzidenzmatrix $M \in \mathbb{R}^{V \times E}$ von G ist definiert als*

$$M_{v,e} = \begin{cases} 1, & \text{falls } v \in e, \\ 0, & \text{sonst.} \end{cases}$$

(b) *Die Adjazenzmatrix $A \in \mathbb{R}^{V \times V}$ von G ist definiert als*

$$A_{v,w} = \begin{cases} 1, & \text{falls } \{v, w\} \in E, \\ 0, & \text{sonst.} \end{cases}$$

(c) *Die Laplacematrix* $L \in \mathbb{R}^{V \times V}$ *von* G *ist definiert als*

$$L_{v,w} = \begin{cases} \deg(v), & \text{falls } v = w, \\ -1, & \text{falls } \{v, w\} \in E, \\ 0, & \text{sonst,} \end{cases}$$

wobei $\deg(v) = |\{w \in V : \{v, w\} \in E\}|$ *der Grad (degree) von* v *ist.*

(d) *Sei* $U \subseteq V$*. Dann ist* $L[U] \in \mathbb{R}^{V \setminus U \times V \setminus U}$ *die Teilmatrix von* L*, die man erhält, wenn man die Zeilen und Spalten streicht, die durch* U *indiziert sind.*

SATZ 1.8. *Sei* $G = (V, E)$ *ein Graph und sei* $v \in V$*. Mit*

$$t(G) = |\{H = (V, T) : H \text{ is Spannbaum von } G\}|$$

bezeichne die Anzahl der Spannbäume von G*. Dann ist*

$$\det L[\{v\}] = t(G).$$

2. Der Algorithmus von Kruskal zur Bestimmung minimaler Spannbäume

Eingabe: $G = (V, E)$ zusammenhängender Graph, $l \in \mathbb{R}^E$ Kantenlängenfunktion
Ausgabe: $H = (V, T)$ ein minimaler Spannbaum von G und l.
Pseudocode:

$T = \emptyset$
while $H = (V, T)$ kein Baum:
  wähle $e \in E \setminus T$ mit $l(e)$ minimal, so dass $H = (V, T \cup \{e\})$ ein Wald ist
  $T = T \cup \{e\}$

DEFINITION 2.1. *Sei* $G = (V, E)$ *ein zusammenhängender Graph und* $l \in \mathbb{R}^E$ *eine Kantenlängenfunktion. Ein Wald* (V, F) *heißt gut (bzgl.* G *und* l*), falls es einen minimalen Spannbaum* (V, T) *von* G *und* l *mit* $F \subseteq T$ *gibt.*

DEFINITION 2.2. *Sei* $G = (V, E)$ *ein Graph. Eine Teilmenge* $C \subseteq E$ *der Kanten heißt ein Schnitt von* G*, falls es eine Knotenmenge* $U \subseteq V$ *mit*

$$C = \delta(U) = \{e \in E : |e \cap U| = 1\}$$

gibt.

SATZ 2.3. *Sei* $G = (V, E)$ *ein zusammenhängender Graph und* $l \in \mathbb{R}^E$ *eine Kantenlängenfunktion. Sei* (V, F) *ein guter Wald und sei* $e \in E \setminus F$ *eine Kante. Falls es einen Schnitt* $C \subseteq E$ *gibt, so dass* $e \in C$ *und* $C \cap F = \emptyset$ *und*

$$l(e) = \min_{e' \in C} l(e')$$

gilt, dann ist $(V, F \cup \{e\})$ *ebenfalls ein guter Wald.*

KOROLLAR 2.4. *Der Algorithmus von Kruskal berechnet einen minimalen Spannbaum.*

3. Matroide und der Greedy-Algorithmus

DEFINITION 3.1. *Sei* X *eine endliche Menge und* $\mathcal{I}$ *eine Menge von Teilmengen von* X*. Das Paar* $(X, \mathcal{I})$ *heißt ein Matroid, falls folgende Bedingungen erfüllt sind:*

(a) $\emptyset \in \mathcal{I}$.
(b) *Falls* $Y \in \mathcal{I}$ *und* $Z \subseteq Y$*, dann ist* $Z \in \mathcal{I}$.
(c) *Falls* $Y, Z \in \mathcal{I}$ *und* $|Y| < |Z|$*, dann gibt es ein* $z \in Z \setminus Y$ *mit* $Y \cup \{z\} \in \mathcal{I}$.

Eine Menge $Y \subseteq X$ heißt unabhängig, falls $Y \in \mathcal{I}$, ansonsten heißt sie abhängig. Sei $Y \subseteq X$. Eine Menge $B \subseteq Y$ heißt eine Basis von Y, falls B eine maximal unabhängige Teilmenge von Y ist. (D.h. für $Z \in \mathcal{I}$ mit $B \subseteq Z \subseteq Y$ gilt $B = Z$.) Eine Basis von X heißt einfach Basis.

BEISPIEL 3.2. *Sei K ein Körper und $M \in K^{m \times n}$ eine Matrix mit Spaltenvektoren $x_1, \ldots, x_n$. Dann ist $(X, \mathcal{I})$, wobei*

$$X = \{1, \ldots, n\} \quad \text{und} \quad \mathcal{I} = \{\{i_1, \ldots, i_k\} \subseteq X : x_{i_1}, \ldots, x_{i_k} \text{ linear unabhängig}\}$$

definiert ist, ein Matroid.

SATZ 3.3. *Ein Paar $(X, \mathcal{I})$ ist genau dann ein Matroid, falls (a), (b) erfüllt sind und die folgende Bedingung gilt:*

(c') *Für alle $Y \subseteq X$ und alle Basen $B_1, B_2 \subseteq Y$ von Y gilt: $|B_1| = |B_2|$.*

BEISPIEL 3.4. *Sei $G = (V, E)$ ein ungerichteter Graph. Dann ist $(E, \mathcal{I})$ ein Matroid, wobei*

$$\mathcal{I} = \{F \subseteq E : (V, F) \text{ ist Wald}\}.$$

DEFINITION 3.5. *Sei X eine endliche Menge und sei $\mathcal{I}$ eine Menge von Teilmengen von X, die die Bedingungen (a) und (b) erfüllt. Sei $w \in \mathbb{R}^X$ eine Gewichtsfunktion. Der Greedy-Algorithmus für $(X, \mathcal{I})$ und w ist definiert als:*

$Y = \emptyset$
while $\exists y \in X : y \notin Y$ und $Y \cup \{y\} \in \mathcal{I}$:
    wähle so ein y mit $w(y)$ maximal
    $Y = Y \cup \{y\}$

SATZ 3.6. *Sei $(X, \mathcal{I})$ ein Paar, das die Bedingungen (a) und (b) erfüllt. Der Greedy-Algorithmus berechnet eine Basis $B \in \mathcal{I}$ mit maximalem Gewicht (d.h. $\sum_{y \in B} w(y) \geq \sum_{y' \in B'} w(y')$ für alle Basen $B' \in \mathcal{I}$) für alle nichnegativen Gewichtsfunktionen $w \in \mathbb{R}^X_{\geq 0}$ genau dann, wenn $(X, \mathcal{I})$ ein Matroid ist.*

KAPITEL 2

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 $V \times V$.*
Eine Kantenfolge *P in D ist ein Tupel der Form*

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

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

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

gilt, d.h. die in P auftretenden Knoten sind alle paarweise verschieden.
Eine Kantenfolge P heißt (gerichteter) Kreis*, falls*

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

gilt.

DEFINITION 1.2. *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*

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

*Ist eine Kantenfolge P ein $s - t$-Weg mit $l(P) = \operatorname{dist}(s, t)$, dann heißt P ein kürzester $s - t$-*Weg*.*

SATZ 1.3. *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 $s - t$-Weg, d.h. $\operatorname{dist}(s, t) < \infty$. Dann existiert eine kürzeste Kantenfolge mit Startknoten s und Endknoten t, die ein Weg ist.*

2. Berechnung kürzester Wege

Input : Gerichteter Graph $D = (V, A)$, $n = |V|$, $s \in V$, $l \in \mathbb{R}^A$
Output : Funktionen $d_0, \ldots, d_n \in (\mathbb{R} \cup \{\infty\})^V$, $g\colon V \setminus \{s\} \to V$
Setze $d_0(s) = 0$.
Setze $d_0(v) = \infty \ \forall v \in V \setminus \{s\}$.
for $k = 0$ ***to*** $n - 1$ **do**
  $d_{k+1}(v) = d_k(v) \ \forall v \in V$
  for $(u, v) \in A$ **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 einen Kreis negativer Länge, der von s aus erreichbar ist“.
end

Algorithmus 1 : Algorithmus von Bellman und Ford

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

$$d_{k+1}(v) = \inf\{l(P) : P \text{ ist } s - v\text{-Kantenfolge, die höchstens } k + 1 \text{ Kanten enthält}\}.$$

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* $\operatorname{dist}(s, v) = d_{n-1}(v)$ *für alle $v \in V$. Dann ist auch*

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

die Umkehrung eines kürzesten $s - v$-Weges.

3. Potentiale

Wie kann man beweisen, dass ein gegebener Weg von s nach t tatsächlich ein kürzester $s - t$-Weg ist? Eine mögliche Beweisstrategie ist es, einfach alle $s - t$-Wege auzuzählen. Diese Strategie wird auch durch 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 3.1. *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 3.2. *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 3.3. *(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 Knoten und angenommen für alle Knoten $v \in V$ gilt $\operatorname{dist}(s, v) < \infty$. Dann gilt die folgende Min-Max-Charakterisierung für die Länge eines kürzesten Weges von s nach t:

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

4. Kürzeste Wege und ganzzahlige lineare Optimierung

DEFINITION 4.1. *Seien $c \in \mathbb{Z}^n$, $b \in \mathbb{Z}^m_{\geq 0}$, $M \in \mathbb{Z}^{m\times n}_{\geq 0}$ gegeben. Ein ganzzahliges lineares Optimierungsproblem (ILP = integer linear program) ist von der Form*

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

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

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

und Kanten für $i = 1, \ldots, n$ 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 $(0, (0, \ldots, 0))$ nach (n, u) mit $u \in U$ genau denen $x \in \mathbb{Z}^n$ mit $x_i \geq 0$ für $i = 1, \ldots, n$ und $(Mx)_j \leq b_j$ für $j = 1, \ldots, m$ mit maximalem Wert $c^\mathsf{T} x$.

KAPITEL 3

Matchings in bipartiten Graphen

1. Berechnung von Matchings mit maximaler Kardinalität

DEFINITION 1.1. *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 $P = (v_0, e_1, v_1, \ldots, e_m, v_m)$ mit der Menge seiner Kanten: Wir schreiben vereinfachend $P = \{e_1, \ldots, e_m\} \subseteq E$.

DEFINITION 1.2. *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 $e_1, \ldots, e_m$ von P sind alternierend nicht aus M und aus M: $e_1 \notin M$, $e_2 \in M$, $e_3 \notin M$, $\ldots$, $e_{m-1} \in M$, $e_m \notin M$.*

LEMMA 1.3. *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) *$M' = M \Delta P = (M \backslash P) \cup (P \backslash M)$ ist ein Matching in M mit $|M'| = |M| + 1$.*

SATZ 1.4. *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.*

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

Nun stellt sich die Frage, wie man einen M-augmentierende Weg findet. Wenn man in den Beweis von Satz 1.4 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 bipartitete 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 1.5. *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:*
$$|e \cap U| = |e \cap W| = 1 \quad \text{für alle } e \in E.$$

DEFINITION 1.6. *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*

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

SATZ 1.7. *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.*

2. 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 2.1. *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 2.2. *(Das Matchingtheorem von 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 2.3. *(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
$$|\Gamma(X)| \geq |X|.$$

KOROLLAR 2.4. *(„Heiratssatz")*
Sei $G = (V, E)$ ein bipartiter Graph. Sei $V = U \cup W$ eine Bipartition. Dann gibt es ein perfektes Matching in M genau dann, wenn $|U| = |W|$ ist und für jede Teilmenge $X \subseteq U$ gilt $|\Gamma(X)| \geq |X|$.

3. Die ungarische Methode

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

DEFINITION 3.1. *Sei $G = (V, E)$ ein Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion auf den Kanten von G.*

(1) *Für $M \subseteq E$ definere 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 3.2. *Sei $G = (V, E)$ ein Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion. Ein Matching $M \subseteq E$ in G heißt* extrem*, falls für alle Matchings $M' \subseteq E$ mit $|M'| = |M|$ gilt, dass $w(M') \leq w(M)$ ist.*

DEFINITION 3.3. *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 $l_M \colon E \to \mathbb{R}$ 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 3.4. *Sei $G = (V, E)$ ein Graph und $w \in \mathbb{R}^E$ eine Gewichtsfunktion. Sei $M \subseteq E$ ein extremes Matching in G. Sei $P \subseteq E$ ein M-augmentierender Weg minimaler Länge. 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 $\max\{w(M_i) : i = 0, 1, \ldots, k-1\}$.

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

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

und mit mit Kantenlängenfunktion $k_M \in \mathbb{R}^{A_M}$ definiert durch

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

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 3.5. *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 k_M keine gerichteten Kreise negativer Länge.*

KAPITEL 4

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 $f \in \mathbb{R}^A_{\geq 0}$ 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*

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

(3) *Ein s-t-Fluss f heißt* beschränkt *durch eine* Kapazitätsfunktion $c \in \mathbb{R}^A_{\geq 0}$, *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 $c \in \mathbb{R}^A_{\geq 0}$. Gesucht ist die Lösung des Maximierungsproblems:*

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

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*

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

(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 $c \in \mathbb{R}^A_{\geq 0}$ eine Kapazitätsfunktion. Dann ist*

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

die Kapazität des Schnittes $\delta^{out}(U)$. 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 $c \in \mathbb{R}^A_{\geq 0}$. Gesucht ist die Lösung des Minimierungsproblems:*

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

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 $c \in \mathbb{R}^A_{\geq 0}$ gegeben. Dann gilt

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

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.

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 Flüsse arbeiten möchte.

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

$$\text{value}(f) \leq c(\delta^{out}(U)) = \sum_{a \in \delta^{out}(U)} c(a) \tag{1}$$

Es gilt Gleichheit in (1) *genau dann, wenn*

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

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

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

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

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

Insbesondere ist f nach Lemma 1.4 maximal.

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

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

Algorithmus zur Bestimmung eines s-t-Flusses mit maximalem Wert von Ford und Fulkerson (1955).

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

SATZ 1.8. *Falls $c(a) \in \mathbb{Q}\ \forall a \in A$, dann terminiert der Ford-Fulkerson Algorithmus in endlich vielen Schritten; sonst im Allgemeinen nicht.*

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

Teil B

Einführung in die konvexe und lineare Optimierung

KAPITEL 5

Konvexität

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 $\langle x, y\rangle = x^\mathsf{T} y$ ist.

DEFINITION 1.1. (1) *Die 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 is*

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

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

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

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

(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 $\lim_{i\to\infty} x_i \in A$.
(2) Die Menge A is 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 $\partial A = \overline{A} \setminus (\operatorname{int} A)$ gelten.

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

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

folgt, dass

$$0 = \alpha_1 = \alpha_2 = \ldots = \alpha_N$$

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*

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

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

Man kann leicht beweisen, dass

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

gilt. Wir werden einen sehr ähnlichen Beweis für die Definition der konvexen Hülle führen.
Affine Unterräume der Dimension 0 sind Punkte. Affine Unterräume der Dimension 1 sind Geraden. Affine Unterräume der Dimension $n-1$ sind Hyperebenen. Man kann eine Hyperebene schreiben als

$$H = \{x \in \mathbb{R}^n : c^{\mathsf{T}} x = \delta\},$$

wobei $c \in \mathbb{R}^n \setminus \{0\}$ und $\delta \in \mathbb{R}$.

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*

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

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.*

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

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

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

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

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

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

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

(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 $y \in \operatorname{conv} A$. Dann gibt es affin unabhängige Punkte $x_1, \dots, x_N \in A$ mit $y \in \operatorname{conv}\{x_1, \dots, x_N\}$. Insbesondere gilt $N \leq \dim A + 1 \leq n + 1$.

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

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

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

SATZ 2.5. *(Ungleichung von Jensen)*
Eine Funktion $f \colon C \to \mathbb{R}$ 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. *Sei $f \colon \mathbb{R}^n \to \mathbb{R}$ 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 $a \in \mathbb{R}^n$ eine positiv semidefinite Matrix ist.

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

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

SATZ 2.8. *Wie oben seien $C \subseteq \mathbb{R}^n$ eine konvexe Menge und $f \colon C \to \mathbb{R}$ 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 Optimum, 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*

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

(2) *Die abgeschlossenen und konvexen Mengen*

$$H^+ = \{x \in \mathbb{R}^n : c^\mathsf{T} x \geq \delta\}$$
$$H^- = \{x \in \mathbb{R}^n : c^\mathsf{T} x \leq \delta\}$$

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 $z \notin C$. 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

$$(z - y)^\mathsf{T}(x - y) \leq 0.$$

Manchmal ist es hilfreich, die Definition von π_C auch auf Punkte $z \in C$ zu erweitern: Wir setzen natürlicherweise $\pi_C(z) = z$.

KOROLLAR 3.4. *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 3.5. *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 3.6. *Die im Beweis des vorhergehenden Satzes konstruierte Trennhyperebene*

$$H = \{x \in \mathbb{R}^n : c^\mathsf{T} x = \delta\} \quad \textit{mit} \quad c = z - \pi_C(z),\ \delta = c^\mathsf{T}\pi_C(z)$$

ist gleichzeitig eine Stützhyperebene von C. Man bekommt eine strikte Trennhyperebene, wenn man $\delta \in (c^\mathsf{T} z, c^\mathsf{T}\pi_C(z))$ wählt.

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*

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

4. Repräsentation konvexer Mengen

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

Innere 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 $\operatorname{ext} C$ bezeichnet.

SATZ 4.3. *(Minkowski; Krein-Milman)*
Sei $C \subseteq \mathbb{R}^n$ eine kompakte und konvexe Menge. Dann gilt $C = \operatorname{conv}(\operatorname{ext}(C))$.

KAPITEL 6

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 endliche Teilmenge $X \subseteq \mathbb{R}^n$ gibt mit*

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

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 $\operatorname{rang} A_z = n$ ist. Dabei ist A_z die Teilmatrix von A die aus den Zeilenvektoren a_j^T von A besteht, für die Gleichheit $a_j^\mathsf{T} z = b_j$ 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*

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

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 $B_n = B(0,1) = \{x \in \mathbb{R}^n : x^\mathsf{T} x \leq 1\}$ die n-dimensionale Einheitskugel ist.*
(4) *Falls $P = \operatorname{conv}\{x_1, \ldots, x_t\}$ ist, dann folgt*
$$P^\times = \{y \in \mathbb{R}^n : x_1^\mathsf{T} y \leq 1, \ldots, x_t^\mathsf{T} y \leq 1\}.$$

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.

2. Das Lemma von Farkas

DEFINITION 2.1. *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*

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

SATZ 2.2. *Seien $x_1, \dots, x_N \in \mathbb{R}^n$. Es gibt eine Matrix $A \in \mathbb{R}^{m \times n}$, so dass*

$$\operatorname{cone}\{x_1, \dots, x_N\} = \{x \in \mathbb{R}^n : Ax \leq 0\}.$$

Insbesondere ist der durch $x_1, \dots, x_N$ erzeugte Kegel ein Polyeder und somit eine abgeschlossene und konvexe Menge.

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

SATZ 2.3. *(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 $A^\mathsf{T} y \geq 0$ und $b^\mathsf{T} y < 0$ erfüllt.

KOROLLAR 2.4. *(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 $y \geq 0$, $A^\mathsf{T} y = 0$ und $b^\mathsf{T} y < 0$ erfüllt.

3. Lineare Optimierung / Lineare Programmierung

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

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

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

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

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*

$$c^\mathsf{T} z \geq c^\mathsf{T} x \quad \textit{für alle } x \in P.$$

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*

$$c^\mathsf{T} x \leq b^\mathsf{T} y.$$

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*

$$(Ax - b)^\mathsf{T} y = 0.$$

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 $j = 1, \ldots, m$.

(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 $(Ax - b)^\mathsf{T} y = 0$.*

(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*

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

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 Umnummerierung

der Zeilen folgende Gestalt:

$$(*) \qquad \begin{bmatrix} 1 & (a_1')^\mathsf{T} \\ \vdots & \vdots \\ 1 & (a_r')^\mathsf{T} \\ -1 & (a_{r+1}')^\mathsf{T} \\ \vdots & \vdots \\ -1 & (a_{r+s}')^\mathsf{T} \\ 0 & (a_{r+s+1}')^\mathsf{T} \\ \vdots & \vdots \\ 0 & (a_m')^\mathsf{T} \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix},$$

wobei $(a_i')^\mathsf{T} \in \mathbb{R}^{1\times(n-1)}$ 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:

$$x_1 + (a_i')^\mathsf{T} \underbrace{\begin{pmatrix} x_2 \\ \vdots \\ x_n \end{pmatrix}}_{=\tilde{x}} \leq b_i \iff x_1 \leq b_i - (a_i')^\mathsf{T}\tilde{x}, \quad i = 1, \ldots, r.$$

Genauso die nächsten s Bedingungen:

$$-x_1 + (a_{r+j}')^\mathsf{T}\tilde{x} \leq b_{r+j} \iff x_1 \geq (a_{r+j}')^\mathsf{T}\tilde{x} - b_{r+j}, \quad j = 1, \ldots, s.$$

Zusammen gilt also

$$(**) \qquad \sup_{j=1,\ldots,s} (a_{r+j}')^\mathsf{T}\tilde{x} - b_{r+j} \leq x_1 \leq \inf_{i=1,\ldots,r} b_i - (a_i')^\mathsf{T}\tilde{x}.$$

(Falls $s = 0$, dann ist $\sup_{j=1,\ldots,s}(a_{r+j}')^\mathsf{T}\tilde{x} - b_{r+j} = -\infty$ und falls $r = 0$, ist $\inf_{i=1,\ldots,r} b_i - (a_i')^\mathsf{T}\tilde{x} = +\infty$. 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

$$\begin{aligned} &(a_{r+j}')^\mathsf{T}\tilde{x} - b_{r+j} \leq b_i - (a_i')^\mathsf{T}\tilde{x}, \quad i = 1, \ldots, r,\ j = 1, \ldots, s \\ &(a_i')^\mathsf{T}\tilde{x} \leq b_i, \quad i = r+s+1, \ldots, m \end{aligned}$$

bzw. das System

$$(***) \qquad \begin{aligned} &((a_{r+j}')^\mathsf{T} + (a_i')^\mathsf{T})\tilde{x} \leq b_i + b_{r+j}, \quad i = 1, \ldots, r,\ j = 1, \ldots, s \\ &(a_i')^\mathsf{T}\tilde{x} \leq b_i, \quad i = r+s+1, \ldots, m. \end{aligned}$$

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* $x_2, x_3, \ldots$ *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

$$\begin{aligned} &\max c^\mathsf{T} x \\ &x \in \mathbb{R}^n \\ &Ax \leq b \end{aligned} \tag{LP}$$

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

$$Ax \leq b,\ \lambda \leq c^\mathsf{T} x.$$

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

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

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 $x_1, \ldots, x_n$ bis λ die letzte Variable ist. Dann wählt man λ so groß wie möglich.

KAPITEL 7

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:*

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

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*

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

DEFINITION 1.3. *Ein Polyeder $P \subseteq \mathbb{R}^n$ heißt* ganzzahlig*, falls für alle $c \in \mathbb{R}^n$, für die $\sup\{c^{\mathsf{T}} x : x \in P\}$ 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-unimoudlaren 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*

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

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:*

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

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 $c \in \mathbb{R}^A_{\geq 0}$ eine Kapazitätsfunktion. Dann gilt

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

Teil C

Algorithmen der linearen Optimierung

KAPITEL 8

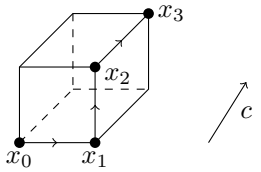
Das Simplexverfahren

Ziel: Löse (LP)

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

Wissen: Angenommen $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ist ein Polytop. Dann wird das Maximum an einer Ecke von P angenommen.
Geometrische Idee: Finde eine Folge von Ecken $x_0, x_1, \ldots, x_N \in P$, so dass

$$c^\mathsf{T} x_0 \leq c^\mathsf{T} x_1 \leq \ldots \leq c^\mathsf{T} x_N = p^*.$$



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 $c^\mathsf{T} = u^\mathsf{T} A$ und $u_i = 0$, falls Zeile i von A nicht zu A_0 gehört. Dazu berechne $c^\mathsf{T} A_0^{-1}$ 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:

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

2.Fall: $u \not\geq 0$.
Sei i der kleinste Index mit $u_i < 0$.

Wähle $y \in \mathbb{R}^n$ mit $a^\mathsf{T} y = 0$ für alle Zeilen a^T von A_0 mit $a^\mathsf{T} \neq a_i^\mathsf{T}$ und $a_i^\mathsf{T} y = -1$.

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

2.a) $a^\mathsf{T} y \leq 0$ für alle Zeilen a^T von A.
Dann ist $x_0 + \lambda y \in P \; \forall \lambda \geq 0$. Desweiteren ist

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

Das heißt das LP ist unbeschränkt.

2.b) $a^\mathsf{T} y > 0$ für eine Zeile a^T von A.
Setze

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

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^T durch Zeile a_j^T 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

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

weil

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

und

$$c^\mathsf{T} y_k = (-u_k)_i > 0.$$

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

$$c^\mathsf{T} x_k = c^\mathsf{T} x_l, \text{ also } x_k = x_{k+1} = \ldots = x_l.$$

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

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

Dann gilt für $j > r$

$$a_j^\mathsf{T} \text{ kommt in } A_p \text{ vor} \Longleftrightarrow a_j^\mathsf{T} \text{ kommt in } A_q \text{ vor.}$$

Es gilt

$$u_p^\mathsf{T} A y_q = c^\mathsf{T} y_q > 0.$$

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

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

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

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

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

Neues LP:

$$\min e^\mathsf{T} y$$
$$\begin{bmatrix} A & -I_m \\ -I_n & 0 \\ 0 & -I_m \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix},$$

mit $e = (1, \ldots, 1)^\mathsf{T}$.

Offensichtliche Ecke:

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

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

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

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

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: $e^\mathsf{T} y^* > 0$.
Dann ist das Orginal-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: $e^\mathsf{T} y^* = 0$.
Dann $y^* = 0$ und x^* ist eine Ecke von P, weil $x^* \in P$ und 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?