

## Gliederung

## Inhaltsverzeichnis

<b>1 Einführung in Lineare Programme und Dualität</b>	<b>1</b>
1.1 Lineare Programme . . . . .	1
1.2 Dualität . . . . .	2
<b>2 Grundlegende Sätze und Definitionen</b>	<b>2</b>
2.1 Starke und schwache Dualität . . . . .	2
2.2 Komplementärer Schlupf . . . . .	3
<b>3 LP und Approximationsalgorithmen am Bsp. des SetCover</b>	<b>4</b>
3.1 Relaxierung eines ganzzahligen linearen Programms . . . . .	4
3.2 Anwendung der Dualität auf den Entwurf von SetCover-Alg. . . . .	6
3.3 Anwendung der Dualität auf die Analyse von SetCover-Alg. . . . .	8

## 1 Einführung in Lineare Programme und Dualität

### Einführung

Als Lineare Programme bezeichnet man Optimierungsprobleme (hier immer Minimum gesucht) mit lin. Zielfunktion. Der Bereich über dem optimiert wird, ist dabei durch eine Menge von linearen Ungleichungsbeschränkungen (Konvexes Polyeder) gegeben. Lineare Programme sind dabei in polynomialer Zeit lösbar. Wichtig ist auch das sogenannte Duale Programm: Ein leicht aus Ausgangsprogramm (Primalprogramm) gewinnbares Programm mit einigen nützlichen Eigenschaften.

### 1.1 Lineare Programme

#### Definition: Lineares Programm

**Definition 1** (Lineares Programm). Ein *Lineares Programm* ist folgendes Optimierungsproblem:

$$\begin{aligned} c^T x &\rightarrow \min! && \text{(Zielfunktion)} \\ \text{subject to } Ax &\geq b && \text{(Zul. Bereich bzw. Nebenbedingungen)} \\ x &\geq 0, \quad x, c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m,n}, m, n \in \mathbb{N} \end{aligned}$$

- Natürlich sind leicht abgewandelte Formen möglich, (z.B. Gleichungsbeschränkungen, Max. statt Min.)
- Aber alle Formen sind äquivalent ineinander umformbar
- Ein Vektor  $x$ , der die Nebenbedingungen erfüllt heißt *zulässig*

## 1.2 Dualität

### Einführendes Beispiel

Beispiel 2 (Primalproblem, [Vaz01]).

$$\begin{aligned} &7x_1 + x_2 + 5x_3 \rightarrow \min! \\ \text{s.t. } &x_1 - x_2 + 3x_3 \geq 10 \\ &5x_1 + 2x_2 - x_3 \geq 6 \\ &x_1, x_2, x_3 \geq 0 \end{aligned}$$

- Wir multiplizieren die beiden Nebenbedingungen mit *positiven*  $y_1, y_2$  und addieren
- Dabei wählen wir  $y_1$  und  $y_2$ , so dass:

Beispiel 3.

$$\begin{aligned} &7x_1 + x_2 + 5x_3 \geq \\ (x_1 - x_2 + 3x_3)y_1 + (5x_1 + 2x_2 - x_3)y_2 &\geq \\ &10y_1 + 6y_2 \end{aligned}$$

### Einführendes Beispiel

Die Idee  $y_1$  und  $y_2$  maximal zu wählen führt auf:

Beispiel 4.

$$\begin{aligned} &10y_1 + 6y_2 \rightarrow \max! \\ \text{s.t. } &y_1 - 5y_2 \leq 7 \\ &5y_1 + 2y_2 \leq 1 \\ &3y_1 - y_2 \leq 5 \\ &y_1, y_2 \geq 0 \end{aligned}$$

Es gilt nun insbesondere:

**Fakt 5.**

$$\max(10y_1 + 6y_2) \leq \min(7x_1 + x_2 + 5x_3)$$

## 2 Grundlegende Sätze und Definitionen

### 2.1 Starke und schwache Dualität

**Allg. Zusammenhang zwischen Primal- und Dualprogramm**

Das zu einem Primalprogramm gehörige Dualprogramm ergibt sich allgemein wie folgt: (Man beachte, dass man das Dualprogramm mit dieser Vorgehensweise völlig “mechanisch” aus dem Ausgangsprgramm gewinnen kann.)

### Primalprogramm

$$\begin{aligned} z(x) &= c^T x \rightarrow \min! \\ \text{s.t. } Ax &\geq b \\ x &\geq 0 \end{aligned}$$

### Dualprogramm

$$\begin{aligned} \vartheta(y) &= b^T y \rightarrow \max! \\ \text{s.t. } A^T y &\leq c \\ y &\geq 0 \end{aligned}$$

### Dualitätssätze

Die oben angesprochenen “nützlichen” Eigenschaften sind die folgenden Dualitätssätze.

**Fakt 6.** für zulässige  $x$  und  $y$  gilt:

$$b^T y \leq c^T x \quad (\text{schwache Dualität})$$

und im Optimum

$$\max b^T y \leq \min c^T x \quad (\text{starke Dualität})$$

## 2.2 Komplementärer Schlupf

Der komplementäre Schlupf beschreibt, welche Bedingungen erfüllt sein müssen, damit die starke Dualität erfüllt ist. Die Gleichheit im Optimum bei Dual und Primal ergibt nämlich sich aus:

**Fakt 7.**

$$\vartheta(y) = b^T y \stackrel{(a)}{\leq} (Ax)^T y = x^T A^T y \stackrel{(b)}{\leq} x^T c = z(x)$$

In Indexschreibweise bedeutet Gleichheit bei (a) und (b):

**Definition 8.** Komplementärer Schlupf

$$y_j > 0 \Rightarrow \text{row}_j Ax = b_j \quad (\text{a})$$

$$x_i > 0 \Rightarrow \text{row}_i A^T y = c_i \quad (\text{b})$$

**Definition 9** (SetCover). Wir werden alle weiteren Überlegungen auf das SetCover-Problem anwenden:

- Eine Probleminstance von SetCover ist eine Menge  $V = \{u_1, \dots, u_n\}$  von Objekten und eine Familie  $\mathcal{S} = \{S_1, \dots, S_m\}$  verschiedener endlicher Mengen aus diesen Objekten.
- Eine Teilfamilie  $\mathcal{S}_{\text{cov}} = \{S_{i_1}, \dots, S_{i_n}\}$  dieser Familie heißt *Überdeckung*, falls  $V(\mathcal{S}_{\text{cov}}) := S_{i_1} \cup \dots \cup S_{i_n} = V$
- Ziel: finde minimale Überdeckung.

**Definition 10** (Kennzahlen). Dabei bezeichne:

- $\text{deg}(u)$  (Grad) die Anzahl der Mengen  $S_j$  in denen  $u$  vorkommt
- $G_{\mathcal{S}}$  die Mächtigkeit der größten Gruppe.
- $\delta_{\mathcal{S}}$  den maximalen Grad.

### 3 LP und Approximationsalgorithmen am Bsp. des Set-Cover

Ein LP für SetCover

$X$ : SetCover LP (ganzzahlig)

$$\begin{aligned} & \sum_{i=1}^m x_i \rightarrow \min! \\ \text{s.t. } & \sum_{i:u \in S_i} x_i \geq 1 \quad \forall u \in V \\ & x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, m\} \end{aligned}$$

Dabei bedeutet  $x_i = 1$ , dass die Menge  $S_i$  in der Überdeckung vorkommt. Die Nebenbedingungen stellen also sicher, dass es für jedes  $u$  mindestens eine Menge in der Überdeckung gibt, die  $u$  enthält.

#### 3.1 Relaxierung eines ganzzahligen linearen Programms

**Relaxierung**

Wir wollen jetzt die Lineare Programmierung nutzen.

Gegeben sei eine Probleminstance  $I$ . Die Idee ist nun einfach die Ganzzahligkeitsbeschränkung zu entfernen. Damit erhalten wir ein relaxiertes Problem  $X_{\text{rel}}$ . Wir bestimmen nun eine Lösung dieses Problems und *gewinnen* daraus ein Lösung  $A(I)$  von  $X$  durch geschicktes Runden. Wir zeigen, dann, dass  $A(I) \leq \rho \text{OPT}(X_{\text{rel}})$  also gilt dann auch:  $A(I) \leq \rho \text{OPT}(X)$ .

## Das relaxierte LP für SetCover

[Wan06]

$X_{\text{rel}}$ : SetCover LP (reell)

$$\begin{aligned} & \sum_{i=1}^m x_i \rightarrow \min! \\ \text{s.t. } & \sum_{i:u \in S_i} x_i \geq 1 \quad \forall u \in V \\ & x_i \in [0, 1] \quad \forall i \in \{1, \dots, m\} \end{aligned}$$

## Ganzzahligkeitslücke

**Definition 11** (Ganzzahligkeitslücke). Bezeichnen  $\mathcal{D}$  die Menge aller Instanzen eines ganzzahlige Optimierungproblems  $X$ . Sei  $X_{\text{rel}}$  das relaxierte Problem. Dann heißt

$$\gamma = \max \left\{ \frac{OPT(X)}{OPT(X_{\text{rel}})} \mid I \in \mathcal{D} \right\}$$

*Ganzzahligkeitslücke.*

**Fakt 12.** Bei einem durch Relaxierung gewonnenen Algorithmus gilt:

$$\rho \geq \gamma$$

*Also beschränkt die Ganzzahligkeitslücke die beweisbare relative Güte beim Einsatz der Relaxierung. Ist die Ganzzahligkeitslücke sehr groß, können wir also mit diesem Ansatz keinen guten Approximationsalgorithmus erreichen.*

## Ganzzahligkeitslücke bei SetCover

[Wan06]

**Satz 13** (Ganzzahligkeitslücke bei SetCover). Für das SetCover-Problem gilt:

$$\gamma \geq \frac{1}{2} \log n$$

*Beweis.* [Skizze,  $\oplus$  = Exklusiv-Oder]

- Betrachte  $V = \{ \bar{u} \mid \bar{u} \in \{0, 1\}^k, u \neq 0^k \}$
- für alle  $\bar{b} \neq 0^k$  als Überdeckungsfamilie:  $S_{\bar{b}} = \{ \bar{u} \mid \bar{u} \oplus u_i \wedge b_i \}$  über GF[2]

□

Wir geben nun einen einfachen deterministischen Algorithmus an, der SetCover approximiert und auf der Relaxation basiert.

### Deterministischer Rundungsalgorithmus

[Wan06]

#### DetRoundSC

Löse LP  $X_{\text{rel}}$

$S_{\text{cov}} := \emptyset$

FOR  $i := 1$  TO  $m$  DO

    IF  $x_i \geq \delta_S^{-1}$  THEN  $S_{\text{cov}} := S_{\text{cov}} \cup \{S_i\}$

#### Satz 14.

Der Algorithmus berechnet tatsächlich eine Überdeckung  $S$  und es gilt:  $\text{DetRoundSC}(S_{\text{cov}}) \leq \delta_S \text{OPT}(S_{\text{cov}})$

Ein weiterer möglicher Ansatz ist das zufällige Runden.

#### Zufälliges Runden

[Wan06]

#### RanRoundSC[r]

Löse LP  $X_{\text{rel}}$

$X := \emptyset$

FOR  $i := 1$  TO  $m$  DO

    Mit Wahrscheinlichkeit  $1 - e^{-rx_i}$  :  $X := X \cup \{S_i\}$

#### Satz 15. Es gilt:

$$\Pr[X \text{ ist Überdeckung}] \geq 1 - ne^{-r} \quad \text{und}$$

$$E[\text{ZF-Wert}] \leq r \text{OPT}(S)$$

Wiederhole nun obigen Algorithmus so lange bis  $X$  tatsächlich eine Überdeckung ist. Es gibt dabei eine (erwartete) obere Schranke für die Anzahl der Wiederholungen.

## 3.2 Anwendung der Dualität auf den Entwurf von SetCover-Alg.

### Dual fitting

Wir wollen nun die Dualität nutzen. Dabei spielt der komplementäre Schlupf eine besondere Rolle. Wir konstruieren im folgenden einen Algorithmus, der Teil (b) ( $x_i > 0 \Rightarrow \text{row}_i A^T y = c_i$ ) zwangsweise erfüllt. Dabei bestimmen wir zunächst eine zulässige Lsg.  $y$  des Duals, die einige Nebenbedingungen scharf macht. Diese muss nicht notwendig optimal sein. Dann setzen wir die zugehörigen  $x_i$  auf 1. Daraus folgt dann:  $\text{row}_i A^T y = c_i x_i$ . Diese Beziehung kann dann zum Bestimmen des Zielfunktionswertes benutzt werden.

Betrachten wir noch einmal die beiden nötigen Programme:

## Erinnerung

**X: SetCover LP (ganzzahlig)**

$$\begin{aligned} & \sum_{i=1}^m x_i \rightarrow \min! \\ \text{s.t. } & \sum_{i:u_j \in S_i} x_i \geq 1 \quad \forall u_j \in V \\ & x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, m\} \end{aligned}$$

**$Y_{rel}$ : relaxiertes Dual für SetCover LP**

$$\begin{aligned} & \sum_{j=1}^n y_j \rightarrow \max! \\ \text{s.t. } & \sum_{j:u_j \in S_i} y_j \geq 1 \quad \forall S_i \in \mathcal{S} \\ & y_i \in [0, 1] \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

## Dual fitting Algorithmus

[Wan06]

### DualPurSC

Löse LP  $Y_{rel}$  Die Lösung sei  $y$

$S_{cov} := \emptyset$

FOR  $i := 1$  TO  $m$  DO

    IF  $i$ -te Nebenbedingung des Duals scharf

        THEN  $x_i := 1$

        ELSE  $x_i := 0$

## Güte des Algorithmus

**Satz 16.** Der Algorithmus bestimmt eine Überdeckung  $\mathcal{S}$  und es gilt:

$$\text{DualPurSC}(\mathcal{S}) \leq \delta_{\mathcal{S}} \text{OPT}(\mathcal{S})$$

*Beweis.* [Skizze] Für alle  $x_i = 1$  gilt  $\sum_{j:u_j \in S_i} y_j = 1 = x_i$  Also gilt:

$$\begin{aligned} \text{DualPurSC}(\mathcal{S}) &= \sum_{i=1}^m x_i = \sum_{i:x_i=1} x_i = \sum_{i:x_i=1} \sum_{j:u_j \in S_i} y_j \leq \\ & \delta_{\mathcal{S}} \sum_{j=1}^n y_j = \delta_{\mathcal{S}} \text{OPT}(Y_{rel}) \leq \delta_{\mathcal{S}} \text{OPT}(\mathcal{S}) \end{aligned}$$

□

### 3.3 Anwendung der Dualität auf die Analyse von SetCover-Alg.

#### Analyse von Algorithmen mit Hilfe der Dualität

Wir berechnen zunächst eine Lsg. des ganzzahligen LPs und konstruieren daraus Lösung  $y$  von  $Y_{\text{rel}}$ . Dann leiten wir eine Beziehung  $c^T x = z(x) = f(y)$  her. ( $x$  sei eine vom Alg. ermittelte Lsg.) Weiter Suchen wir nach einer Beziehung  $f(y) \leq g^T(y) = b^T y$ . Dann gilt schließlich:  $z(x) \leq g^{\text{OPT}}(I)$  ( $I$  ist eine Instanz des ganzzahligen Optimierungsproblems)

#### Analysealgorithmus

[Wan06] Sei  $\mathcal{H}(n) := \sum_{i=1}^n \frac{1}{i}$

#### DualPurSC

FOR  $i := 1$  TO  $m$  DO  $x_i := 0$

$C := \emptyset$  (schon gespeicherte Objekte)

WHILE  $C \neq V$  DO

    Bestimme einen Index  $i$  mit  $|S_i \setminus C|$  maximal

$x_i = 1$

    FOR alle  $u_j \in S_i \setminus C$  DO

$\text{preis}(u_j) := \frac{1}{|S_i \setminus C|}$

$y_j := \frac{1}{\mathcal{H}(G_S)} \text{preis}(u_j)$

$C := C \cup S_i$

**Satz 17.** *Es gilt:  $\text{PrimalDualSC}(S) \leq \mathcal{H}(G_S) \text{OPT}(S)$*

Man beachte, dass die innere FOR-Schleife, die das  $y$  konstruiert für den eigentlichen Algorithmus unnötig ist, und nur der Analyse dient.

#### Beweis mit Hilfe der Dualität

*Beweis.* [Skizze] Es gilt zunächst, dass die Lsg.  $y$  des Alg. tatsächlich eine Lsg. des Duals ist. (nicht offensichtlich) Also:

$$\begin{aligned} \text{PrimalDualSC}(S) &= \sum_{i=1}^m x_i = \sum_{i: x_i=1} x_i = \sum_{i: x_i=1} \sum_{u_j \in S_i \setminus C} \text{preis}(u_j) = \\ &= \sum_{j=1}^n \text{preis}(u_j) = \mathcal{H}(G_S) \sum_{j=1}^n y_j \leq \mathcal{H}(G_S) \text{OPT}(S) \end{aligned}$$

□

## **Literatur**

[Vaz01] V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[Wan06] Rolf Wanka. *Approximationsalgorithmen - Eine Einführung*. Teubner, 2006.