



Technische  
Universität  
Braunschweig



# Mathematische Methoden der Algorithmik – Vorlesung #04

Arne Schmidt

# Letzte Woche



# Dualität

Maximiere $c^T x$ Unter Bedingungen $Ax \leq b$ $x \geq 0$	
Minimiere $b^T y$ Unter Bedingungen $A^T y \geq c$ $y \geq 0$	-Maximiere $-b^T y$ Unter Bedingungen $-A^T y \leq -c$ $y \geq 0$

## Theorem (Starke Dualität)

Besitzt das primale LP eine optimale Lösung  $x^*$ , dann besitzt auch das duale LP eine optimale Lösung  $y^*$  und es gilt

$$c^T x^* = b^T y^*$$

## Theorem (Komplementärer Schlupf)

Angenommen,  $x = (x_1, \dots, x_n)$  und  $y = (y_1, \dots, y_m)$  sind gültige Lösungen für das primale bzw. duale LP. Dann sind diese genau dann optimal, wenn:

1.  $\forall j = 1, \dots, n: x_j z_j = 0$
2.  $\forall i = 1, \dots, m: y_i w_i = 0$

Wobei  $w = (w_1, \dots, w_m)$  und  $z = (z_1, \dots, z_n)$  die zugehörigen primalen bzw. dualen Schlupfvariablen sind.



# Dualer Simplex



# Primales Dictionary infeasible, aber dual feasible?

Wenn das primale Dictionary infeasible ist, haben wir eine Phase I gestartet. Aber ist das immer notwendig?

$\zeta =$	0 -	$1x_1 -$	$1x_2$
$w_1 =$	4 +	$2x_1 +$	$1x_2$
$w_2 =$	-8 +	$2x_1 -$	$4x_2$
$w_3 =$	-7 +	$1x_1 -$	$3x_2$

Infeasible Dictionary!

$-\xi =$	0 -	$4y_1 +$	$8y_2 +$	$7y_3$
$z_1 =$	1 -	$2y_1 -$	$2y_2 -$	$1y_3$
$z_2 =$	1 -	$1y_1 +$	$4y_2 +$	$3y_3$

Feasible Dictionary!

Wir können Simplex direkt über das duale Dictionary durchführen!



# Dual pivotisieren I

Betrachte komplementären Schlupf: Wenn wir bspw.  $y_2$  in die Basis aufnehmen, muss  $w_2$  den Wert 0 annehmen, geht also aus der Basis heraus!

Durch negativ-transponierte Matrix A: Nimm Basisvariable heraus, die die kleinste Konstante besitzt.

$$\begin{array}{rcccc} \zeta = & 0 - & 1x_1 - & 1x_2 \\ \hline w_1 = & 4 + & 2x_1 + & 1x_2 \\ w_2 = & -8 + & 2x_1 - & 4x_2 \\ w_3 = & -7 + & 1x_1 - & 3x_2 \end{array}$$

Infeasible Dictionary!

$$\begin{array}{rcccc} -\xi = & 0 - & 4y_1 + & 8y_2 + & 7y_3 \\ \hline z_1 = & 1 - & 2y_1 - & 2y_2 - & 1y_3 \\ z_2 = & 1 - & 1y_1 + & 4y_2 + & 3y_3 \end{array}$$

Feasible Dictionary!



# Dual pivotisieren II

Wenn  $w_2$  heraus geht, wer geht in die Basis rein?

Über das duale erhalten wir  $z_1$  oder  $z_2$ . Da einer der beiden auf 0 geht, darf das entsprechende  $x_1$  bzw  $x_2$  größer werden!

Hier: Tausche  $y_2$  mit  $z_1$  im dualen Dictionary, also tausche  $w_2$  mit  $x_1$  im primalen Dictionary!

$$\begin{array}{rcccc} \zeta = & 0 - & 1x_1 - & 1x_2 \\ \hline w_1 = & 4 + & 2x_1 + & 1x_2 \\ w_2 = & -8 + & 2x_1 - & 4x_2 \\ w_3 = & -7 + & 1x_1 - & 3x_2 \end{array}$$

Infeasible Dictionary!

$$\begin{array}{rcccc} -\xi = & 0 - & 4y_1 + & 8y_2 + & 7y_3 \\ \hline z_1 = & 1 - & 2y_1 - & 2y_2 - & 1y_3 \\ z_2 = & 1 - & 1y_1 + & 4y_2 + & 3y_3 \end{array}$$

Feasible Dictionary!

Größter negativer Quotient  $\left(\frac{a_{ij}}{c_j}\right)$



# Dual pivotisieren III

Worüber pivotisieren wir nun?

$w_3$  muss raus,  $w_2$  kommt dafür rein!

$$\begin{array}{rcccc} \zeta = & -4 & - & .5w_2 & - & 3x_2 \\ \hline w_1 = & 12 & + & 1w_2 & + & 5x_2 \\ x_1 = & 4 & + & .5w_2 & + & 2x_2 \\ w_3 = & \boxed{-3} & + & \boxed{.5w_2} & - & 1x_2 \end{array}$$

Infeasible Dictionary!

$$\begin{array}{rcccc} -\xi = & 4 & - & 12y_1 & - & 4z_1 + & 3y_3 \\ \hline y_2 = & .5 & - & 1y_1 & - & .5z_1 & - & .5y_3 \\ z_2 = & 3 & - & 5y_1 & - & 2z_1 & + & 1y_3 \end{array}$$

Feasible Dictionary!



# Dual pivotisieren IV

Nun sind beide Dictionaries feasible und wir haben sogar eine optimale Lösung!

$$\begin{array}{rcccc} \zeta = & -7 - & 1w_3 - & 4x_2 \\ \hline w_1 = & 18 + & 2w_3 + & 7x_2 \\ x_1 = & 7 + & 1w_3 + & 3x_2 \\ w_2 = & 6 + & 2w_3 + & 2x_2 \end{array}$$

Feasible Dictionary!

$$\begin{array}{rcccc} -\xi = & 7 - & 18y_1 - & 7z_1 - & 6y_3 \\ \hline y_2 = & 1 - & 2y_1 - & 1z_1 - & 2y_3 \\ z_2 = & 4 - & 7y_1 - & 3z_1 - & 2y_3 \end{array}$$

Feasible Dictionary!



# Dual pivotisieren

Ganz allgemein können wir das primale Dictionary folgendermaßen dual pivotisieren, falls negative  $b_i$  existieren.

- Wähle **Basisvariable** nach der kleinsten negativen Konstante.  
(Falls keine Variable mehr negativ ist, ist das Dictionary optimal!)
- Wähle **Nichtbasisvariable** mit dem größten negierten Quotienten  $\left(\frac{a_{ij}}{c_j}\right)$



# Duale Phase I?



# Duale Phase I!

$\zeta =$	$0 -$	$1x_1 +$	$4x_2$
$w_1 =$	$4 +$	$2x_1 +$	$1x_2$
$w_2 =$	$-8 +$	$2x_1 -$	$4x_2$
$w_3 =$	$-7 +$	$1x_1 -$	$3x_2$

$-\xi =$	$0 -$	$4y_1 +$	$8y_2 +$	$7y_3$
$z_1 =$	$1 -$	$2y_1 -$	$2y_2 -$	$1y_3$
$z_2 =$	$-4 -$	$1y_1 +$	$4y_2 +$	$3y_3$

Suche eine Zielfunktion, sodass das duale Dictionary feasible wird. Z.B.:  $-x_1 - x_2$

$\eta =$	$0 -$	$1x_1 -$	$1x_2$
$w_1 =$	$4 +$	$2x_1 +$	$1x_2$
$w_2 =$	$-8 +$	$2x_1 -$	$4x_2$
$w_3 =$	$-7 +$	$1x_1 -$	$3x_2$

$-\mu =$	$0 -$	$4y_1 +$	$8y_2 +$	$7y_3$
$z_1 =$	$1 -$	$2y_1 -$	$2y_2 -$	$1y_3$
$z_2 =$	$1 -$	$1y_1 +$	$4y_2 +$	$3y_3$

Nun dualen Simplex auf das primale Dictionary anwenden!



# Duale Phase I – Ergebnis

$\eta =$	$-7 -$	$1w_3 -$	$4x_2$
$w_1 =$	$18 +$	$2w_3 +$	$7x_2$
$x_1 =$	$7 +$	$1w_3 +$	$3x_2$
$w_2 =$	$6 +$	$2w_3 +$	$2x_2$

Mit  $\zeta = -x_1 + 4x_2 = -(7 + w_3 + 3x_2) + 4x_2$  können wir ein Dictionary zum originalen LP wieder herstellen.

$\zeta =$	$-7 -$	$1w_3 +$	$1x_2$
$w_1 =$	$18 +$	$2w_3 +$	$7x_2$
$x_1 =$	$7 +$	$1w_3 +$	$3x_2$
$w_2 =$	$6 +$	$2w_3 +$	$2x_2$

Wir sehen direkt: Das LP ist unbeschränkt!



# Duale Phase I – Infeasibility



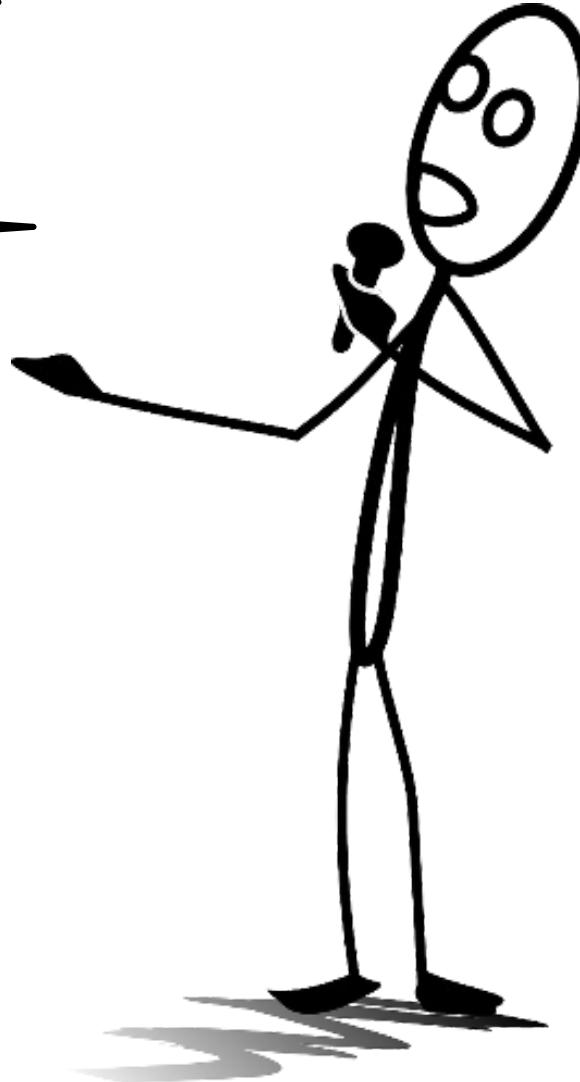
# Recap

Jedes LP besitzt ein duales LP.

Sind beide feasible, besitzen sie denselben optimalen Lösungswert.

Über komplementären Schlupf können wir die duale Lösung bestimmen.

Wenn Dictionary dual feasible ist, können wir den dualen Simplex-Algorithmus durchführen



# Dualisieren von LP im Allgemeinen



# Dualisieren im Allgemeinen – Gleichungen

Betrachte eine Gleichung  $a_i x = b_i$ , was äquivalent ist zu

1.  $a_i x \leq b_i$ , sowie
2.  $-a_i x \leq -b_i$

Damit werden im dualen zwei Variablen  $y_i^+ \geq 0$  und  $y_i^- \geq 0$  erzeugt.

Wie treten diese Variablen im dualen LP in der Zielfunktion und in den Constraints auf?

Sie müssen immer in der Form  $ay_i^+ - ay_i^-$  für einen Koeffizienten  $a$  vorkommen!

Damit können wir aber auch eine freie Variable  $y_i$  benutzen, die positiv oder negativ sein kann!

Also: Eine Gleichung  $a_i x = b_i$  erzeugt eine freie Variable  $y_i$ .

Auch andersherum: Eine freie Variable  $y_i$  erzeugt eine Gleichung  $a_i x = b_i$ .



# Dualisieren im Allgemeinen – Ungleichung

Betrachte eine Gleichung  $a_i x \geq b_i$ , was äquivalent ist zu  $-a_i x \leq -b_i$

Damit wird im dualen eine Variable  $y_i \geq 0$  erzeugt.

Achte bei den Constraints und der Zielfunktion auf das Vorzeichen!

Empfohlen: Erst in  $-a_i x \leq -b_i$  umwandeln!

Also: Eine Gleichung  $a_i x \geq b_i$  erzeugt eine Variable  $y_i \geq 0$ .

Auch andersherum: Eine Variable  $y_i \geq 0$  erzeugt eine Ungleichung  $a_i x \leq b_i$  (passe auf Vorzeichen auf).



# Beispiel – Dualisieren

$$\max 3x_1 + 4x_2 - 2x_3$$

s.t.

$$-x_1 - x_2 - 3x_3 \geq 4$$

$$2x_1 + 3x_2 - 2x_3 \leq 10$$

$$-3x_1 + 2x_2 - x_3 \geq 2$$

$$x_1, x_2 \geq 0$$

$x_3$  frei

$$\max 3x_1 + 4x_2 - 2x_3$$

s.t.

$$x_1 + x_2 + 3x_3 \leq -4$$

$$2x_1 + 3x_2 - 2x_3 \leq 10$$

$$3x_1 - 2x_2 + x_3 \leq -2$$

$$x_1, x_2 \geq 0$$

$x_3$  frei

$$\min -4y_1 + 10y_2 - 2y_3$$

s.t.

$$y_1 + 2y_2 + 3y_3 \geq 3$$

$$y_1 + 3y_2 - 2y_3 \geq 4$$

$$3y_1 - 2y_2 + 1y_3 = -2$$

$$y_1, y_2, y_3 \geq 0$$



# Ausblick: Integer Programming



# Ein ganzzahliges Programm (IP)

Betrachte einen Graphen  $G = (V, E)$ .

$$\begin{aligned} \max \quad & \sum_{e \in E} x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} x_e \leq 1, \quad \forall v \in V \\ & x_e \in \{0,1\}, \quad \forall e \in E \end{aligned}$$

Probleme:

- Wir können Simplex nicht darauf anwenden.  
Warum?
- Wir können das duale Programm nicht direkt bestimmen.

Schauen wir uns das *relaxierte* Problem an.

Dieses IP modelliert das  
Matching-Problem



# Ein relaxiertes ganzzahliges Programm (IP)

Betrachte einen Graphen  $G = (V, E)$ .

$$\begin{aligned} & \max \sum_{e \in E} x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} x_e \leq 1, \quad \forall v \in V \\ & x_e \geq 0, \quad \forall e \in E \end{aligned}$$

Fragen:

- Warum benötigen wir die Bedingung  $x_e \leq 1$  nicht?
- Was ist das duale LP dazu?

Dieses LP modelliert das  
*fraktionale Matching-*  
Problem



# Ein relaxiertes ganzzahliges Programm (IP)

Betrachte einen Graphen  $G = (V, E)$ .

$$\begin{aligned} & \max \sum_{e \in E} x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} x_e \leq 1, \quad \forall v \in V \\ & x_e \geq 0, \quad \forall e \in E \end{aligned}$$

$$\begin{aligned} & \min \sum_{v \in V} y_v \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} y_v \geq 1, \quad \forall e \in E \\ & y_v \geq 0, \quad \forall v \in V \end{aligned}$$

Dieses LP modelliert das  
*fraktionale Matching-*  
Problem

Welches Problem ist das?



# Ein relaxiertes ganzzahliges Programm (IP)

Betrachte einen Graphen  $G = (V, E)$ .

$$\max \sum_{e \in E} x_e$$

s.t.

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in V$$
$$x_e \in \{0,1\}, \forall e \in E$$

$\text{IP}_M$

$$\max \sum_{e \in E} x_e$$

s.t.

$$\sum_{e \in \delta(v)} x_e \leq 1, \forall v \in V$$
$$x_e \geq 0, \forall e \in E$$

$\text{LP}_M$

$$\min \sum_{v \in V} y_v$$

s.t.

$$y_v + y_w \geq 1, \forall \{v, w\} \in E$$
$$y_v \geq 0, \forall v \in V$$

$\text{LP}_{VC}$

$$\min \sum_{v \in V} y_v$$

s.t.

$$y_v + y_w \geq 1, \forall \{v, w\} \in E$$
$$y_v \in \{0,1\}, \forall v \in V$$

$\text{IP}_{VC}$

Wie stehen  $\text{IP}_M$ ,  $\text{LP}_M$ ,  $\text{LP}_{VC}$  und  $\text{IP}_{VC}$  zueinander?

Klar:  $\text{LP}_M = \text{LP}_{VC}$



# Relaxierte IPs

Wird ein IP relaxiert, d.h. Ganzzahlige Variablen dürfen reelle Werte annehmen, kann der optimale Zielfunktionswert nicht schlechter werden.

Damit gilt also  $\text{IP}_M \leq \text{LP}_M = \text{LP}_{VC} \leq \text{IP}_{VC}$

Beim Unterschied zwischen IP und LP spricht man auch vom **Integrality Gap**.

Wie groß kann dieser Unterschied werden?

Man kann sich schnell davon überzeugen, dass  $\text{IP}_M \leq 2 \cdot \text{IP}_{VC}$  gilt. Also ist der Integrality Gap sowohl für Matching als auch für Vertex Cover maximal 2.



# Ausblick



# Was noch kommt...

