Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems

Institute for Informatics

Approximationsalgorithmen

Lineare Programmierung
und
LP-Dualität

4. Vorlesung

Philipp Kindermann

Wintersemester 2019/20

Einführung Lineare Programmierung

Ein Großteil bekannter Approximationsalgorithmen basiert auf linearer Programmierung.

- Lineare Programmierung (LP) und LP-Dualität
- Min-Max-Beziehungen
- Techniken LP-basierten Algorithmendesigns

Motivation: Obere und untere Schranken

Betrachte schweres NP-Minimierungsproblem.

Entscheidungsproblem:
Ist gegebenes S **obere Schranke** für OPT?

Leicht verifizierbare „Ja“-Zertifikate (nach Def. von NP)

Untere Schranken / „Nein“-Zertifikate?
$\rightsquigarrow$ wahrscheinlich nicht! (Vermutung: NP $\neq$ coNP)

Brauchen untere Schranken $\mathrm{obj}(S) \geq \mathrm{OPT}/\alpha$
(approximative „Nein“-Zertifikate) für
Approximationsalgorithmen!

Beispiele

- Vertex Cover: Untere Schranke durch Matchings
- TSP: Untere Schranke durch MST oder Cycle Cover

Lineare Programmierung

Optimiere (d.h. minimiere oder maximiere) eine lineare Funktion unter Anwesenheit linearer Ungleichungsbedingungen.

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \text{Standardform (ÜA)} \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

Beispiel. $c = \begin{pmatrix} 7 \\ 1 \\ 5 \end{pmatrix} \quad A = \begin{pmatrix} 1 & -1 & 3 \\ 5 & 2 & -1 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 6 \end{pmatrix}$

$$\begin{array}{lrcrcrcr} \textbf{minimize} & 7x_1 & + & x_2 & + & 5x_3 & & \\ \textbf{subject to} & x_1 & - & x_2 & + & 3x_3 & \geq & 10 \\ & 5x_1 & + & 2x_2 & - & x_3 & \geq & 6 \\ & & & & & x_1, x_2, x_3 & \geq & 0 \end{array}$$

Lin. Progr. – Obere Schranken

Optimiere (d.h. minimiere oder maximiere) eine lineare Funktion unter Anwesenheit linearer Ungleichungsbedingungen.

$$
\begin{array}{lrcrcrcr}
\textbf{minimize} & 7x_1 14 + & & x_2 1 & + & 5x_3 15 & = & 30 \\
\textbf{subject to} & x_1 2 & - & x_2 1 & + & 3x_3 9 & \geq & 10 10 \\
 & 5x_1 10 + & & 2x_2 2 & - & x_3 3 & \geq & 6 9 \\
 & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

Zulässige Lösung?

$x = (2, 1, 3)$

$\Rightarrow \text{obj}(x) = 30$ ist obere Schranke für OPT

Lin. Progr. – Untere Schranken

Optimiere (d.h. minimiere oder maximiere) eine lineare Funktion unter Anwesenheit linearer Ungleichungsbedingungen.

$$
\begin{array}{lrcrcrcr}
\textbf{minimize} & 7x_1 & + & x_2 & + & 5x_3 & & \\
\textbf{subject to} & x_1 & - & x_2 & + & 3x_3 & \geq & 10 \\
 & 5x_1 & + & 2x_2 & - & x_3 & \geq & 6 \\
 & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

$$7x_1 + x_2 + 5x_3 \geq x_1 - x_2 + 3x_3 \Rightarrow \text{OPT} \geq 10$$

$$\begin{aligned}
7x_1 + x_2 + 5x_3 &\geq (x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3) \\
&\geq 10 + 6 \qquad \Rightarrow \text{OPT} \geq 16
\end{aligned}$$

Lin. Progr. – Untere Schranken

Primal

$$
\begin{array}{lrcrcrcr}
\textbf{minimize} & 7x_1 & + & x_2 & + & 5x_3 & & \\
\textbf{subject to} & y_1(x_1 & - & x_2 & + & 3x_3) & \geq & 10\,y_1 \\
 & y_2(5x_1 & + & 2x_2 & - & x_3) & \geq & 6\,y_2 \\
 & & & & x_1, x_2, x_3 & & \geq & 0
\end{array}
$$

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

Dual

$$
\begin{array}{lrcrcr}
\textbf{maximize} & 10y_1 & + & 6y_2 & & \\
\textbf{subject to} & y_1 & + & 5y_2 & \leq & 7 \\
 & -y_1 & + & 2y_2 & \leq & 1 \\
 & 3y_1 & - & y_2 & \leq & 5 \\
 & & & y_1, y_2 & \geq & 0
\end{array}
$$

Jede zulässige Lösung für das **duale** Programm liefert untere Schranke für Optimum des **primalen** Programms.

$x = (7/4, 0, 11/4)$ und $y = (2, 1)$ liefern beide den Wert $26 = \mathrm{OPT}$.

Primal – Dual

Primales Programm

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

Duales Programm

$$\begin{array}{lrcl} \textbf{maximize} & b^\top y & & \\ \textbf{subject to} & A^\top y & \leq & c \\ & y & \geq & 0 \end{array}$$

Duales Programm des Dualen Programmes

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

LP-Dualität

Primal

$$\begin{array}{lrcl} \textbf{minimize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

Dual

$$\begin{array}{lrcl} \textbf{maximize} & b^\intercal y & & \\ \textbf{subject to} & A^\intercal y & \leq & c \\ & y & \geq & 0 \end{array}$$

Satz. Das primale Programm hat ein endliches Optimum $\Leftrightarrow$ das duale Programm hat ein endliches Optimum. Sind dann $x^* = (x_1^*, \ldots, x_n^*)$ und $y^* = (y_1^*, \ldots, y_m^*)$ *optimale* Lösungen für das primale bzw. duale Programm, so gilt

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^* .$$

Schwache LP-Dualität

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

$$\begin{array}{lrcl} \textbf{maximize} & b^\top y & & \\ \textbf{subject to} & A^\top y & \leq & c \\ & y & \geq & 0 \end{array}$$

Satz. Sind $x = (x_1, \ldots, x_n)$ und $y = (y_1, \ldots, y_m)$ *zulässige* Lsg. für das primale bzw. duale Programm, so gilt

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Beweis.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \overset{\geq}{\geq} \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i\right)} x_j = \sum_{i=1}^{m} \boxed{\left(\sum_{j=1}^{n} a_{ij} x_j\right)} y_i \overset{\geq}{\geq} \sum_{i=1}^{m} \boxed{b_i} y_i \,.$$

Komplementärer Schlupf (*complementary slackness*)

minimize	$c^\top x$		
subject to	Ax	$\geq$	b
	x	$\geq$	0

maximize	$b^\top y$		
subject to	$A^\top y$	$\leq$	c
	y	$\geq$	0

Satz. Seien $x = (x_1, \ldots, x_n)$ und $y = (y_1, \ldots, y_m)$ zulässige Lösungen für das primale bzw. duale Programm. Dann sind x und y genau dann optimal, wenn die folgenden Bedingungen erfüllt sind:

Primaler KS:
Für jedes $j = 1, \ldots, n$: Entweder $x_j = 0$ oder $\sum_{i=1}^m a_{ij} y_i = c_j$

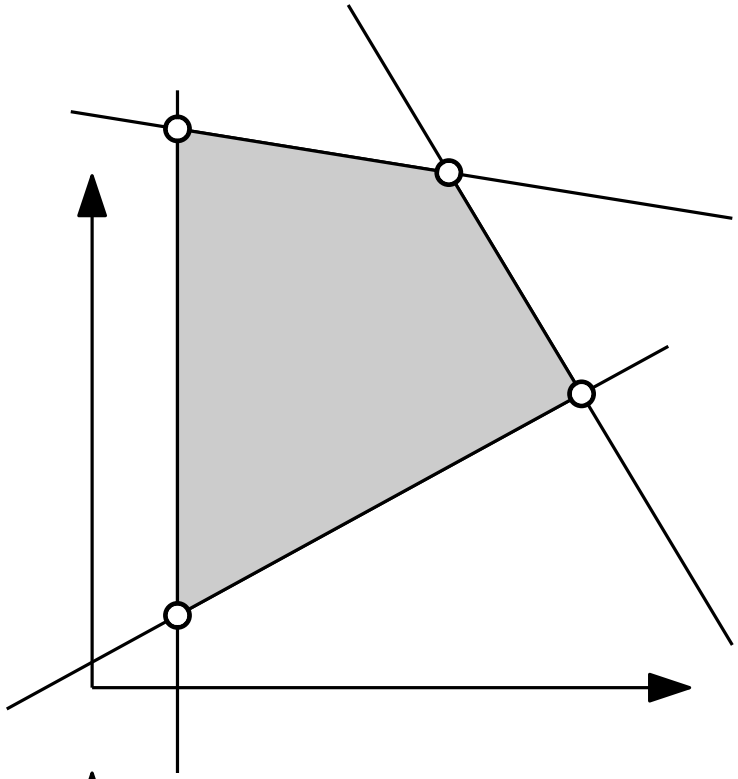
Dualer KS:
Für jedes $i = 1, \ldots, m$: Entweder $y_i = 0$ oder $\sum_{j=1}^n a_{ij} x_j = b_i$

Beweis. Folgt aus LP-Dualität:

$$\sum_{j=1}^n \boxed{c_j} x_j \geq \sum_{j=1}^n \boxed{\left(\sum_{i=1}^m a_{ij} y_i\right)} x_j = \sum_{i=1}^m \boxed{\left(\sum_{j=1}^n a_{ij} x_j\right)} y_i \geq \sum_{i=1}^m \boxed{b_i} y_i \,.$$

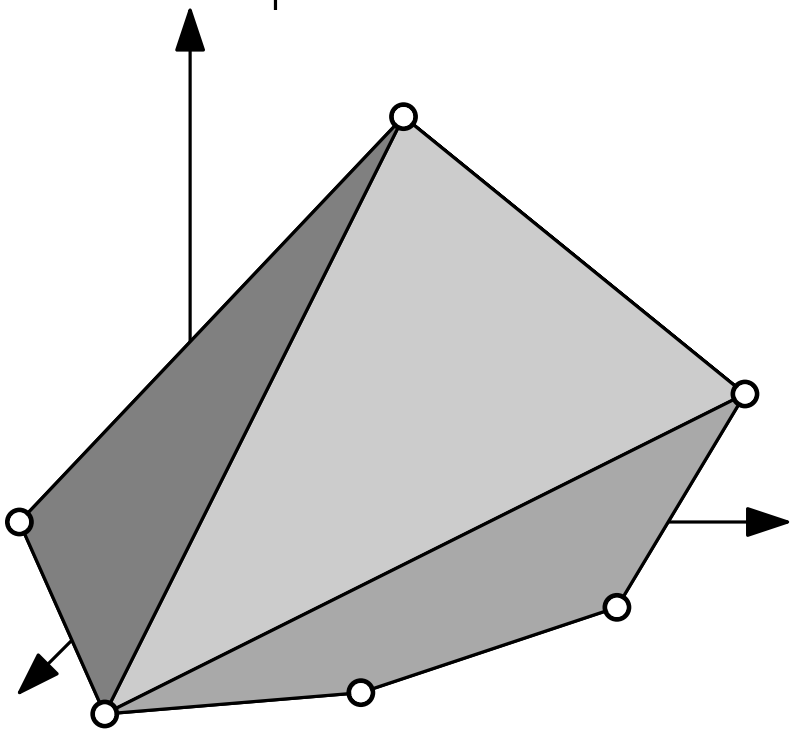
LP und konvexe Polytope

Menge der zulässigen Lösungen eines LPs mit n Variablen bilden **konvexes Polytop** im $\mathbb{R}^n$ (Schnitt von Halbräumen).



Ecken des Polytops heißen **Extrempunkt-Lösungen** $\Leftrightarrow$ n linear unabhängige Ungleichungen sind mit Gleichheit erfüllt.

Wenn optimale Lösung existiert, dann auch eine solche, die Extrempunkt ist.



Ganzzahlige lineare Programme

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \in & \mathbb{N} \end{array}$$

Eine Vielzahl von NP-Optimierungsproblemen lässt sich als ILP formulieren; Lösung ist daher NP-schwer.

LP-Relaxierung liefert untere Schranke: $\mathrm{OPT}_{\mathrm{LP}} \leq \mathrm{OPT}_{\mathrm{ILP}}$

Einführung Lineare Programmierung

Ein Großteil bekannter Approximationsalgorithmen basiert auf linearer Programmierung.

- Lineare Programmierung (LP) und LP-Dualität
- Min-Max-Beziehungen
- Techniken LP-basierten Algorithmendesigns

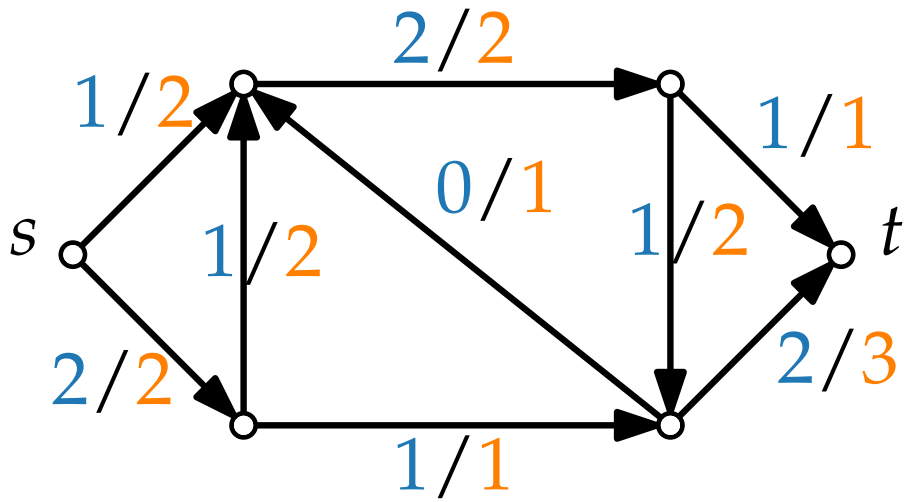
Max-Flow-Problem

Gegeben: Ein gerichteter Graph $G = (V, E)$ mit Kantenkapazitäten $c\colon E \to \mathbb{Q}_+$ und zwei ausgezeichnete Knoten Quelle s und Senke t.

Gesucht: Ein maximaler s–t-Fluss (nicht-negative Kantengewichte f), so dass

- $f(u,v) \leq c(u,v)$ für jede Kante $(u,v) \in E$
- $\sum_{(u,v)\in E} f(u,v) = \sum_{(v,z)\in E} f(v,z)$ für alle $v \in V - \{s,t\}$

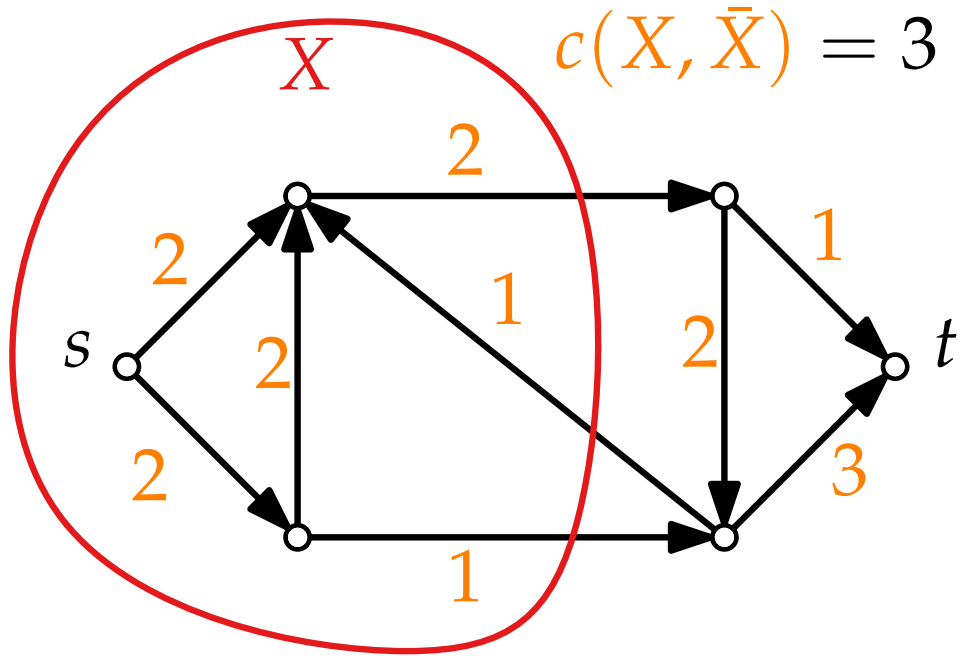
Wert des Flusses hierbei Zufluss zu t minus Abfluss aus t



Min-Cut-Problem

Gegeben: Ein gerichteter Graph $G = (V, E)$ mit Kantenkapazitäten $c\colon E \to \mathbb{Q}_+$ und zwei ausgezeichnete Knoten Quelle s und Senke t.

Gesucht: Ein s–t-Schnitt, d.h. eine Knotenmenge X mit $s \in X$ und $t \in \overline{X}$, so dass das Gesamtgewicht $c(X, \overline{X})$ der Kanten von X nach $\overline{X}$ minimal ist.

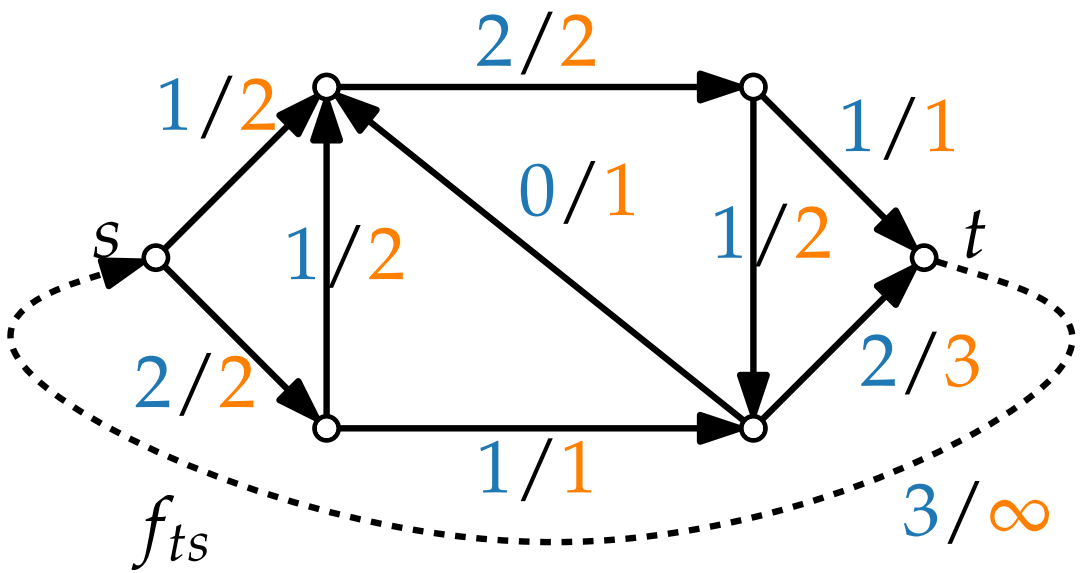


Max-Flow-Min-Cut-Theorem

Satz. Der Wert eines maximalen s–t-Flusses und die Kapazität eines minimalen s–t-Schnitts sind gleich.

Beweis. Spezialfall des LP-Dualitätssatzes...

maximize	f_{ts}	
subject to	$f_{uv} \leq c_{uv}$	$\forall (u,v) \in E \setminus \{(t,s)\}$
	$\sum_{u\colon (u,v)\in E} f_{uv} - \sum_{z\colon (v,z)\in E} f_{vz} \leq 0$	$\forall v \in V$
	$f_{uv} \geq 0$	$\forall (u,v) \in E$



Max-Flow-Min-Cut-Theorem

Satz. Der Wert eines maximalen s–t-Flusses und die Kapazität eines minimalen s–t-Schnitts sind gleich.

Beweis. Spezialfall des LP-Dualitätssatzes...

$$
\begin{array}{lll}
\textbf{maximize} & f_{ts} & \\
\textbf{subject to} & f_{uv} \leq c_{uv} & \forall (u,v) \in E \setminus \{(t,s)\} \quad d_{uv} \\
& \displaystyle\sum_{u\colon (u,v)\in E} f_{uv} - \sum_{z\colon (v,z)\in E} f_{vz} \leq 0 & \forall v \in V \quad p_v \\
& f_{uv} \geq 0 & \forall (u,v) \in E
\end{array}
$$

maximize $c^\intercal x \;= \sum_{(u,v)\in E} (0 \cdot f_{uv}) \;+1 \cdot f_{ts} \;\Rightarrow c = (0,\ldots,0,1)$

In welchen Nebenbedingungen kommt $f_{uv} \neq f_{ts}$ vor? d_{uv}, p_u, p_v

$$\Rightarrow d_{uv} - p_u + p_v \geq 0$$

In welchen Nebenbedingungen kommt f_{ts} vor? p_s, p_t

$$\Rightarrow p_s - p_t \geq 1$$

Max-Flow-Min-Cut-Theorem

Satz. Der Wert eines maximalen s–t-Flusses und die Kapazität eines minimalen s–t-Schnitts sind gleich.

Beweis. Spezialfall des LP-Dualitätssatzes...

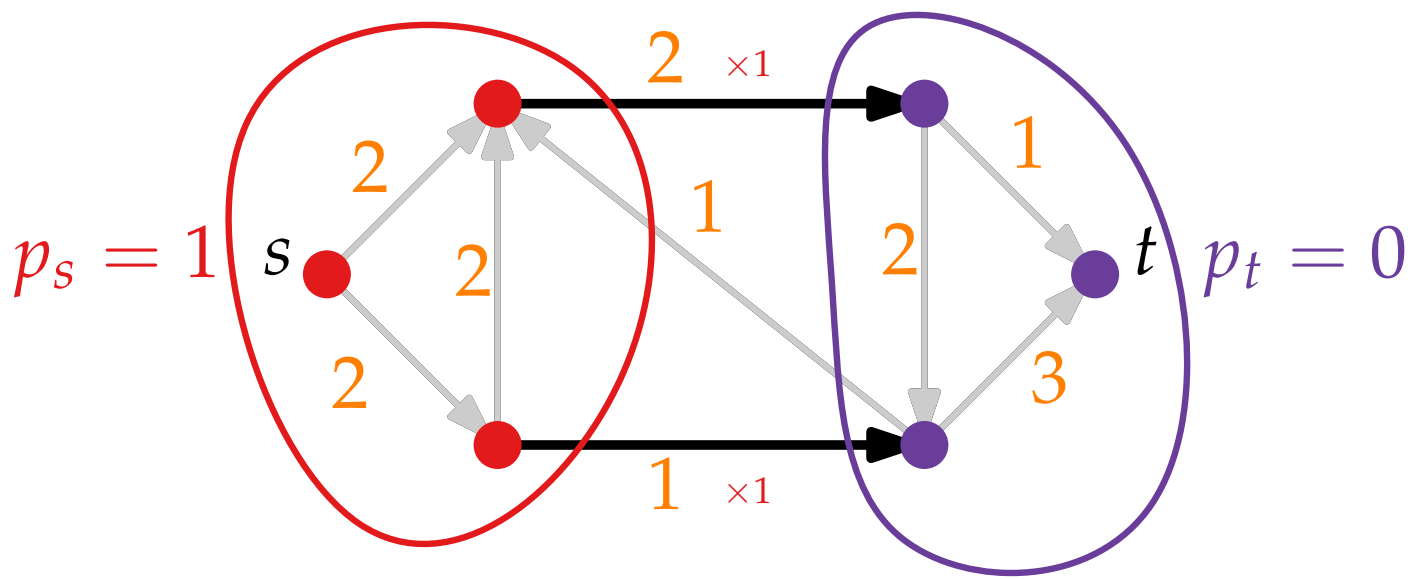
$$
\begin{array}{llll}
\textbf{maximize} & f_{ts} & & \\
\textbf{subject to} & f_{uv} \leq c_{uv} & \forall (u,v) \in E \setminus \{(t,s)\} & d_{uv} \\
& \displaystyle\sum_{u\colon (u,v)\in E} f_{uv} - \sum_{z\colon (v,z)\in E} f_{vz} \leq 0 & \forall v \in V & p_v \\
& f_{uv} \geq 0 & \forall (u,v) \in E &
\end{array}
$$

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{(u,v)\in E\setminus\{(t,s)\}} c_{uv} \cdot d_{uv} & \\
\textbf{subject to} & d_{uv} - p_u + p_v \geq 0 & \forall (u,v) \in E \setminus \{(t,s)\} \\
& p_s - p_t \geq 1 & \\
& d_{uv} \geq 0 & \forall (u,v) \in E \\
& p_u \geq 0 & \forall u \in V
\end{array}
$$

Duales LP – Interpretation als ILP

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{(u,v)\in E\setminus\{(t,s)\}} c_{uv}\cdot d_{uv} & \\
\textbf{subject to} & d_{uv} - p_u + p_v \geq 0 & \forall (u,v)\in E\setminus\{(t,s)\} \\
 & p_s - p_t \geq 1 & \\
 & d_{uv} \cancel{\geq 0} \in \{0,1\} & \forall (u,v)\in E \\
 & p_u \cancel{\geq 0} \in \{0,1\} & \forall u \in V
\end{array}
$$

äquivalent zu Min-Cut!



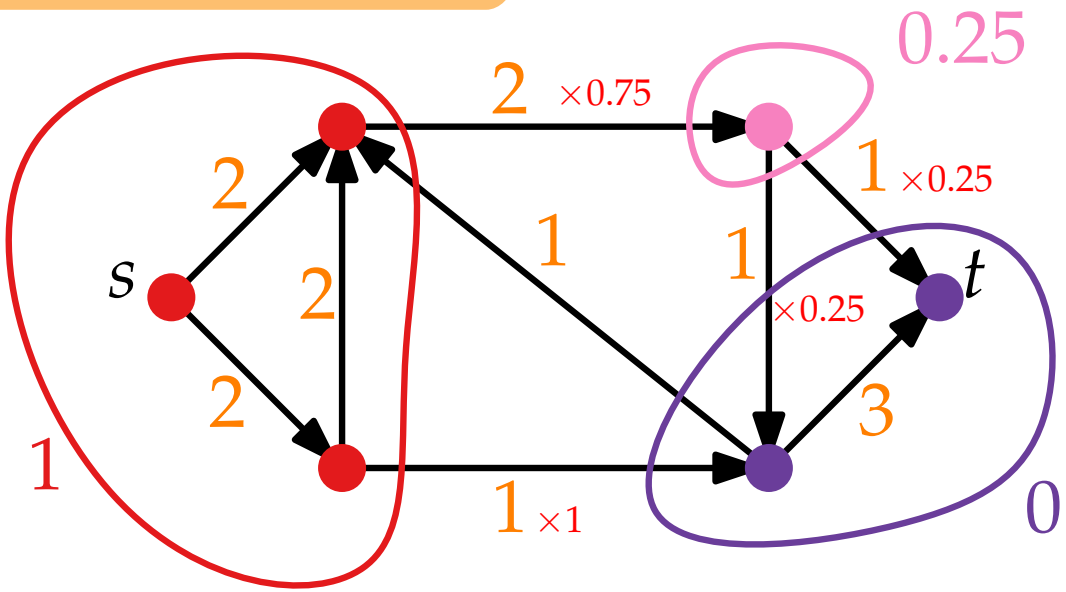
Duales LP – Fraktionale Schnitte

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{(u,v)\in E\setminus\{(t,s)\}} c_{uv}\cdot d_{uv} & \equiv \text{LP-Relaxierung des ILPs} \\ \textbf{subject to} & d_{uv} - p_u + p_v \geq 0 & \forall (u,v)\in E\setminus\{(t,s)\} \\ & p_s - p_t \geq 1 & \\ & d_{uv} \geq 0 & \forall (u,v)\in E \\ & p_u \geq 0 & \forall u \in V \end{array}$$

Jede Extrempunkt-Lösung ist hier **ganzzahlig**! (ÜA)

Jeder s–t-Weg $s = v_0, \ldots, v_k = t$ hat Länge ≥ 1 bezüglich d:

$$\sum_{i=0}^{k-1} d_{i,i+1} \geq \sum_{i=0}^{k-1} (p_i - p_{i+1})$$
$$= p_s - p_t$$



Duales LP – Komplementärer Schlupf

maximize	f_{ts}	
subject to	$f_{uv} \leq c_{uv}$	$\forall (u,v) \in E \setminus \{(t,s)\}$
	$\sum_{u\colon (u,v)\in E} f_{uv} - \sum_{z\colon (v,z)\in E} f_{vz} \leq 0$	$\forall v \in V$
	$f_{uv} \geq 0$	$\forall (u,v) \in E$

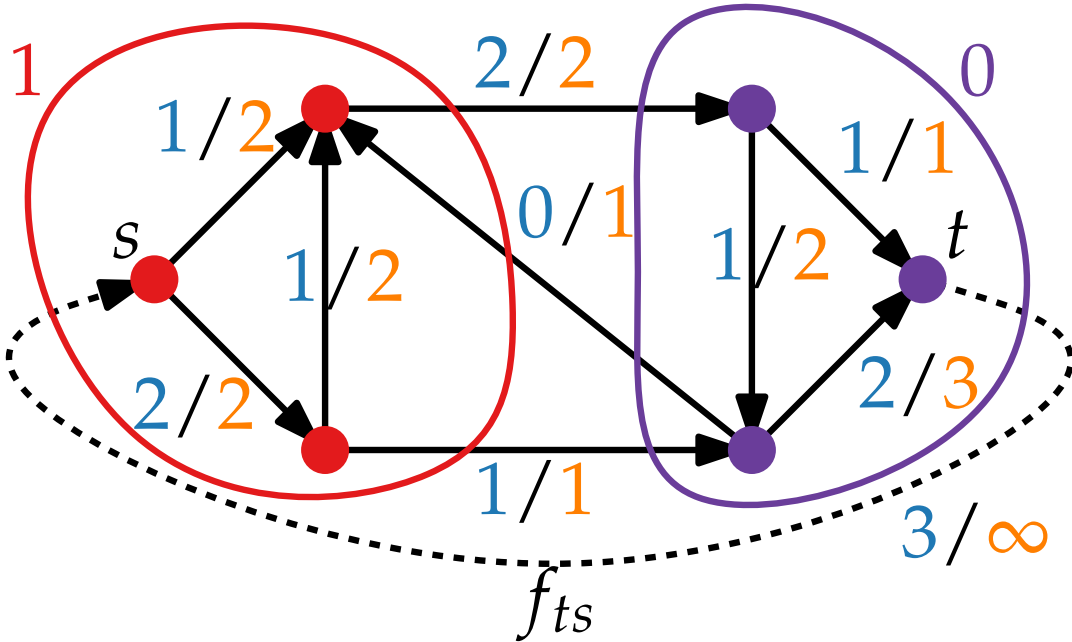
minimize	$\sum_{(u,v)\in E\setminus\{(t,s)\}} c_{uv} \cdot d_{uv}$	
subject to	$d_{uv} - p_u + p_v \geq 0$	
	$p_s - p_t \geq 1$	
	$d_{uv} \geq 0$	
	$p_u \geq 0$	

Primaler KS:
$\forall j$: Entweder $x_j = 0$ oder $\sum_{i=1}^{m} a_{ij} y_i = c_j$

Dualer KS:
$\forall i$: Entweder $y_i = 0$ oder $\sum_{j=1}^{n} a_{ij} x_j = b_i$

Für einen maximalen Fluss und minimalen Schnitt gilt:

- Für Vorwärtskanten (u,v) des Schnitts gilt $f_{uv} = c_{uv}$.
  ($d_{uv} = 1$, also nach dualem KS: $f_{uv} = c_{uv}$.)
- Für Rückwärtskanten (u,v) des Schnitts gilt $f_{uv} = 0$.
  (Sonst wäre nach primalem KS: $d_{uv} - 0 + 1 = 0$.)



Einführung Lineare Programmierung

Ein Großteil bekannter Approximationsalgorithmen basiert auf linearer Programmierung.

- Lineare Programmierung (LP) und LP-Dualität
- Min-Max-Beziehungen
- Techniken LP-basierten Algorithmendesigns

I) LP-Runden

Betrachte Minimierungsproblem Π in ILP-Form

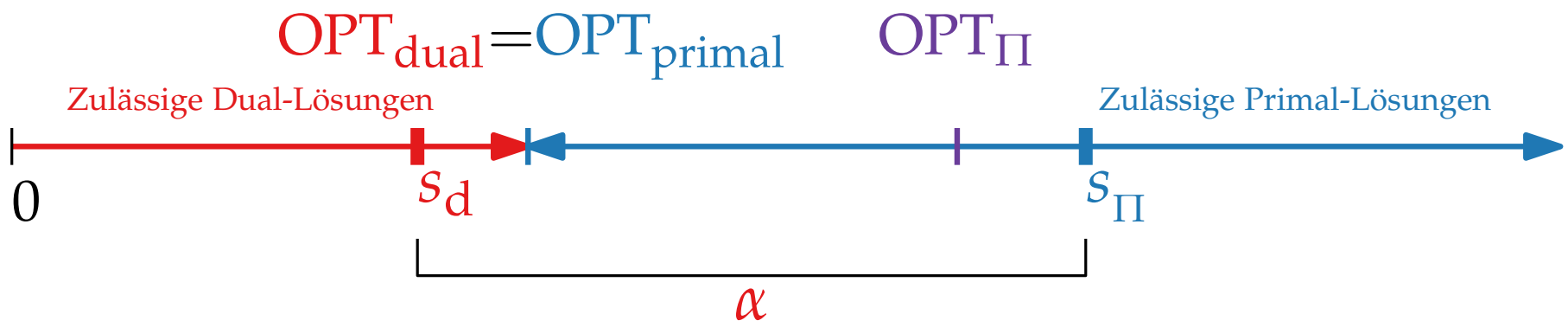
Berechne Lösung der LP-Relaxierung

Runde zu ganzzahliger Lösung für Π

Schwierigkeit: **Zulässige** Lösung für Π

Approximationsgüte $\mathrm{ALG}/\mathrm{OPT}_{\Pi} \leq \mathrm{ALG}/\mathrm{OPT}_{\mathrm{relax}} \leq \alpha$

II) Primal-Dual-Ansatz



Betrachte Minimierungsproblem Π in ILP-Form.

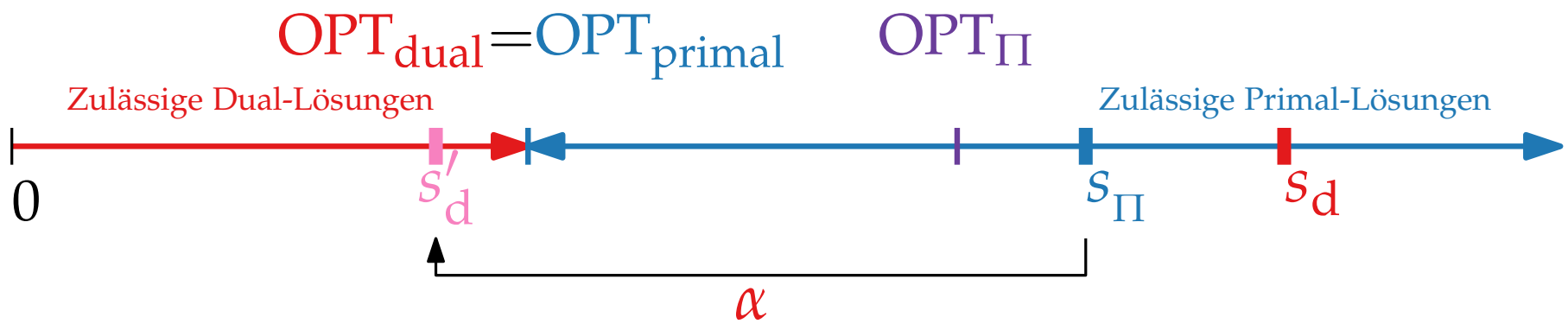
Starte mit (trivialer) zulässiger Duallösung und unzulässiger Primallösung (z.B. alle Variable $= 0$).

Berechne **duale** Lösung s_{d} und ganzzahlige Lösung s_{Π} für Π
iterativ: erhöhe s_{d} entspr. KS und mache s_{Π} „ganzzahliger“.

Approximationsgüte $\leq \mathrm{obj}(s_{\Pi}) / \mathrm{obj}(s_{\mathrm{d}})$

Vorteil:
Kommt ohne LP-„Maschinerie“ aus; ggf. schneller, flexibler.

III) Dual Fitting



Betrachte Minimierungsproblem Π in ILP-Form.

Kombinatorischer Algorithmus berechnet zul. Lösung s_{Π} und „unzulässige“ Duallösung s_{d}, die s_{Π} „voll auszahlt“,

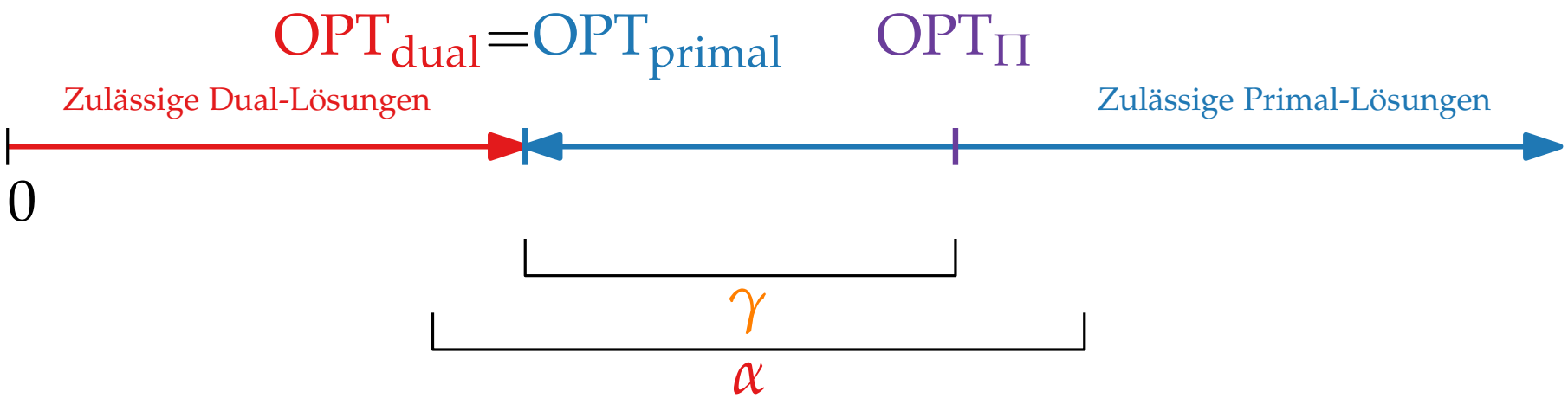
d.h. $\mathrm{obj}(s_{\Pi}) \leq \mathrm{obj}(s_{\mathrm{d}})$.

Skaliere Dualvariablen, so dass neue Duallösung s'_{d} *zulässig*.

$\Rightarrow \mathrm{obj}(s_{\Pi})/\alpha = \mathrm{obj}(s'_{\mathrm{d}}) \leq \mathrm{OPT}_{\mathrm{dual}} \leq \mathrm{OPT}_{\Pi}$

$\Rightarrow$ Skalierungsfaktor α ist Approximationsfaktor.

Wichtig: Integrality Gap



Betrachte Minimierungsproblem Π in ILP-Form.

Ohne Ausnutzung weiterer Eigenschaften sind alle diese Methoden höchstens so gut wie das *Integrality Gap* der LP-Relaxierung:

$$\alpha \geq \gamma = \sup_{I} \frac{\mathrm{OPT}_{\Pi}(I)}{\mathrm{OPT}_{\mathrm{primal}}(}$$