

Graphen und diskrete Optimierung

im Bachelorstudiengang 'Informatik und Nachhaltigkeit'

Lineare Programmierung - Fortsetzung

Marie Schmidt

21.06.2023

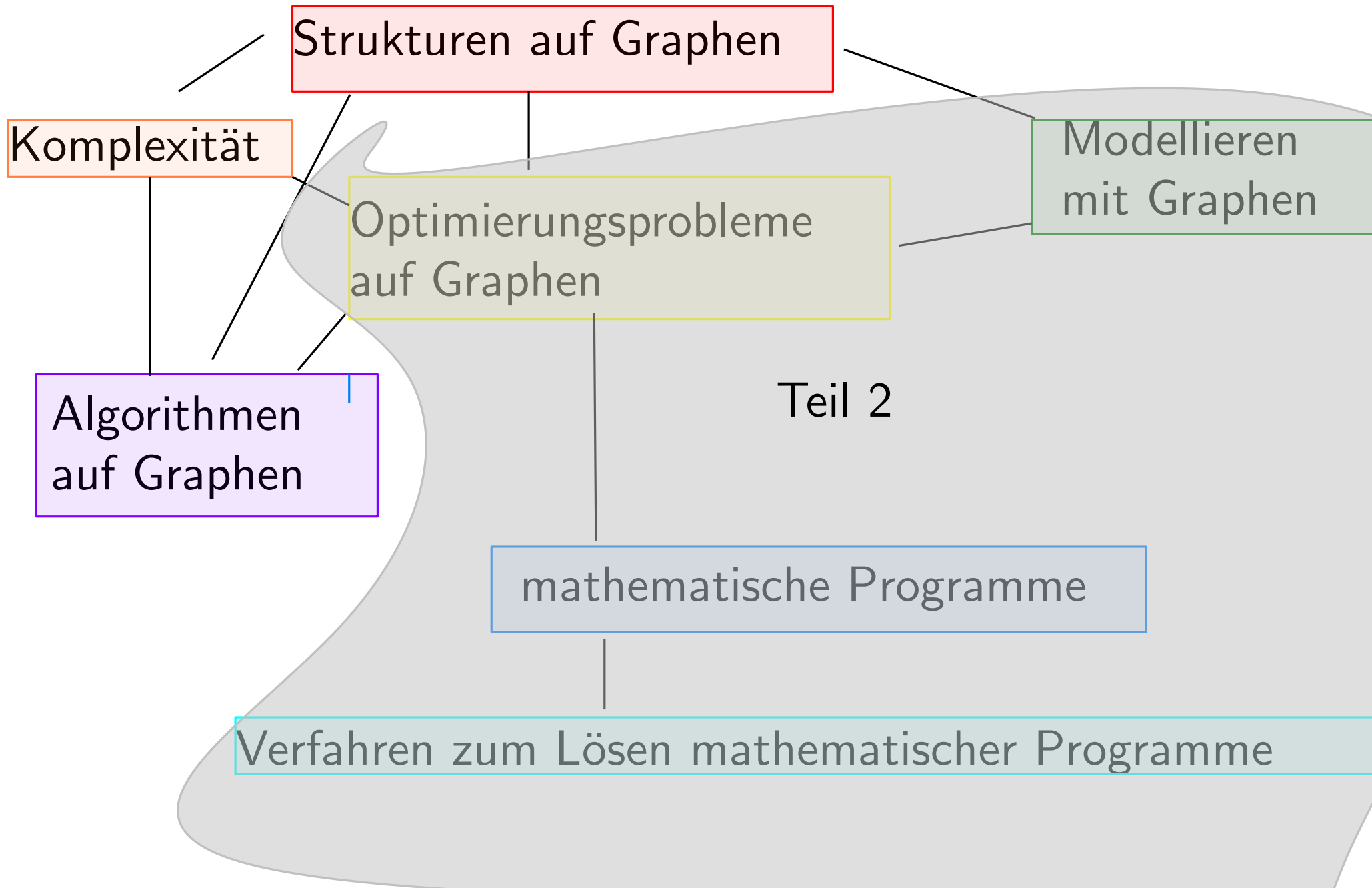
Mündliche Prüfungen am 26. und 28.7.

Bitte Termin über Doodle wählen (first-come, first-served)

<https://tinyurl.com/GudO2023>



Worum soll er hier gehen?



Vorlesungsübersicht

- Einführung: Modellieren auf Graphen ✓
- Teil 1: Graphen und Algorithmen auf Graphen ✓
- Teil 2: Mathematische Programmierung
 - Einführung in die lineare und ganzzahlig lineare Programmierung ✓
 - * Was sind lineare / ganzzahlig lineare Programme ✓
 - * Modellierung als lineares / ganzzahlig lineares Programm ✓
 - * Graphische Lösungsmethode für lineare Programme mit zwei Variablen ✓
 - lineare Programmierung
 - * Hauptsatz der linearen Optimierung ✓
 - * Simplexverfahren
 - * Komplexität lineare Programmierung
 - * Flussprobleme als lineare Programme
 - * Dualität
 - ganzzahlige Programmierung

Zusammengefasst: Algorithmus Simplexverfahren

Require: Optimierungsproblem (P) der Form $\max\{x : Ax = b, x \geq 0\}$

Ensure: optimale Basislösung von (P)

Finde zulässige Basislösung $x^{(0)}$ von (P) .

Setze $i := 0$.

while keine Optimallösung gefunden wurde **do**

Überprüfe anhand der Zielfunktion, ob sich $x^{(0)}$ durch Aufnehmen einer der Nicht-Basisvariablen in die Basis erhöhen lässt.

if Zielfunktion lässt sich nicht verbessern **then**

return $x^{(i)}$ ist optimale Basislösung

else if Zielfunktion lässt sich durch Aufnahme von x_j verbessern **then**

Überprüfe, um wieviel x_j maximal erhöht werden darf, ohne die Nebenbedingungen zu verletzen,

Erhöhe x_j um diesen Betrag,

Erstelle $LP^{(i+1)}$ durch Umformen der Nebenbedingungen: löse die beschränkende Nebenbedingungen nach x_j auf setze sie in die restlichen Nebenbedingungen und in die Zielfunktion ein.

An $LP^{(i+1)}$ lässt sich die aktuelle zulässige Basislösung und der aktuelle Wert der Zielfunktion ablesen.


end if

end while

Zusammengefasst: Algorithmus Simplexverfahren

Require: Optimierungsproblem (P) der Form $\max\{x : Ax = b, x \geq 0\}$

Ensure: optimale Basislösung von (P)

Finde zulässige Basislösung $x^{(0)}$ von (P) .  **Wie machen wir das?**

Setze $i := 0$.

while keine Optimallösung gefunden wurde **do**

Überprüfe anhand der Zielfunktion, ob sich $x^{(0)}$ durch Aufnehmen einer der Nicht-Basisvariablen in die Basis erhöhen lässt.

if Zielfunktion lässt sich nicht verbessern **then**

return $x^{(i)}$ ist optimale Basislösung

else if Zielfunktion lässt sich durch Aufnahme von x_j verbessern **then**

Überprüfe, um wieviel x_j maximal erhöht werden darf, ohne die Nebenbedingungen zu verletzen,

Erhöhe x_j um diesen Betrag,

Erstelle $LP^{(i+1)}$ durch Umformen der Nebenbedingungen: löse die beschränkende Nebenbedingungen nach x_j auf setze sie in die restlichen Nebenbedingungen und in die Zielfunktion ein.

An $LP^{(i+1)}$ lässt sich die aktuelle zulässige Basislösung und der aktuelle Wert der Zielfunktion ablesen.

end if

end while

Finden einer zulässigen Basislösung

Hat man ein LP (wie im Produktionsplanungsbeispiel) in der Form

$$\begin{array}{lll}
 \max & c_1 x_1 & + c_2 x_2 + \dots + c_n x_n \\
 \text{so dass} & a_{11} x_1 & + a_{12} x_2 + \dots + a_{1n} x_n \leq b_1 \\
 & a_{21} x_1 & + a_{22} x_2 + \dots + a_{2n} x_n \leq b_2 \\
 & \dots & \\
 & a_{m1} x_1 & + a_{m2} x_2 + \dots + a_{mn} x_n \leq b_m \\
 & x_1, & x_2, \dots, x_n \geq 0
 \end{array}$$

mit
 $b_i \geq 0$
 für alle
 i

Finden einer zulässigen Basislösung

Hat man ein LP (wie im Produktionsplanungsbeispiel) in der Form

$$\begin{array}{lll}
 \max & c_1 x_1 & + c_2 x_2 + \dots + c_n x_n \\
 \text{so dass} & a_{11} x_1 & + a_{12} x_2 + \dots + a_{1n} x_n \leq b_1 \\
 & a_{21} x_1 & + a_{22} x_2 + \dots + a_{2n} x_n \leq b_2 \\
 & \dots & \\
 & a_{m1} x_1 & + a_{m2} x_2 + \dots + a_{mn} x_n \leq b_m \\
 & x_1, & x_2, \dots, x_n \geq 0
 \end{array}$$

mit
 $b_i \geq 0$
 für alle
 i

Achtung: auch diese Form wird
 in der Literatur mitunter
 'Standardform' oder
 'Normalform' genannt

Finden einer zulässigen Basislösung

Hat man ein LP (wie im Produktionsplanungsbeispiel) in der Form

$$\begin{array}{llll}
 \max & c_1 x_1 & + c_2 x_2 + \dots & + c_n x_n \\
 \text{so dass} & a_{11} x_1 & + a_{12} x_2 + \dots & + a_{1n} x_n \leq b_1 \\
 & a_{21} x_1 & + a_{22} x_2 + \dots & + a_{2n} x_n \leq b_2 \\
 & \dots & & \\
 & a_{m1} x_1 & + a_{m2} x_2 + \dots & + a_{mn} x_n \leq b_m \\
 & x_1, & x_2, \dots & x_n \geq 0
 \end{array}$$

mit
 $b_i \geq 0$
 für alle
 i

dann kann man nach Einführen der Schlupfvariablen

$$\begin{array}{llllll}
 \max & c_1 x_1 & + c_2 x_2 + \dots & + c_n x_n & & \\
 \text{so dass} & a_{11} x_1 & + a_{12} x_2 + \dots & + a_{1n} x_n + y_1 & & = b_1 \\
 & a_{21} x_1 & + a_{22} x_2 + \dots & + a_{2n} x_n & + y_2 & = b_2 \\
 & \dots & & & & \\
 & a_{m1} x_1 & + a_{m2} x_2 + \dots & + a_{mn} x_n & + y_m & = b_m \\
 & x_1, x_2, & \dots x_n, y_1, y_2, & \dots, y_m & \geq 0 &
 \end{array}$$

folgende Basislösung direkt ablesen:

Finden einer zulässigen Basislösung

Hat man ein LP (wie im Produktionsplanungsbeispiel) in der Form

$$\begin{array}{llll}
 \max & c_1 x_1 & + c_2 x_2 + \dots & + c_n x_n \\
 \text{so dass} & a_{11} x_1 & + a_{12} x_2 + \dots & + a_{1n} x_n \leq b_1 \\
 & a_{21} x_1 & + a_{22} x_2 + \dots & + a_{2n} x_n \leq b_2 \\
 & \dots & & \\
 & a_{m1} x_1 & + a_{m2} x_2 + \dots & + a_{mn} x_n \leq b_m \\
 & x_1, & x_2, \dots & x_n \geq 0
 \end{array}$$

mit
 $b_i \geq 0$
 für alle
 i

dann kann man nach Einführen der Schlupfvariablen

$$\begin{array}{llllll}
 \max & c_1 x_1 & + c_2 x_2 + \dots & + c_n x_n & & \\
 \text{so dass} & a_{11} x_1 & + a_{12} x_2 + \dots & + a_{1n} x_n + y_1 & & = b_1 \\
 & a_{21} x_1 & + a_{22} x_2 + \dots & + a_{2n} x_n & + y_2 & = b_2 \\
 & \dots & & & & \\
 & a_{m1} x_1 & + a_{m2} x_2 + \dots & + a_{mn} x_n & + y_m & = b_m \\
 & x_1, x_2, & \dots x_n, y_1, y_2, & \dots, y_m & \geq 0 &
 \end{array}$$

folgende Basislösung direkt ablesen: $y_1 = b_1, y_2 = b_2, \dots, y_m = b_m$

Finden einer zulässigen Basislösung

Warum geht das eigentlich nicht für jedes beliebige LP?

(Optimierungsbeispiel) in der Form

$$\begin{aligned}
 & \dots + c_n x_n \\
 \text{so dass } & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\
 & \dots \\
 & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\
 & x_1, x_2, \dots, x_n \geq 0
 \end{aligned}$$

mit
 $b_i \geq 0$
 für alle
 i

dann kann man nach Einführen der Schlupfvariablen

$$\begin{aligned}
 & \max \quad c_1x_1 + c_2x_2 + \dots + c_nx_n \\
 \text{so dass } & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + y_1 = b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + y_2 = b_2 \\
 & \dots \\
 & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + y_m = b_m \\
 & x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m \geq 0
 \end{aligned}$$

folgende Basislösung direkt ablesen: $y_1 = b_1, y_2 = b_2, \dots, y_m = b_m$

Finden einer zulässigen Basislösung

ansonsten:

Finden einer zulässigen Basislösung

ansonsten:

Möglichkeit 1: Bringe LP auf gewünschte Form (siehe vorherige Slide)
bringen

Finden einer zulässigen Basislösung

ansonsten:

Möglichkeit 2: **2-Phasen-Simplex**: Löse Hilfsproblem (für das zulässige Basislösung bekannt ist) mit Simplexmethode um zulässige Basislg für ursprüngliches Problem zu finden.

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

- Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist:
multipliziere Zeile mit -1

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

- Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist:
multipliziere Zeile mit -1
- Füge Schlupf-/Überschussvariablen ein, um Gleichheitsform (mit positiver rechter Seite) zu bekommen.

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

- Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist:
multipliziere Zeile mit -1
- Füge Schlupf-/Überschussvariablen ein, um Gleichheitsform (mit positiver rechter Seite) zu bekommen.
- Wo noch keine *Schlupf*variable ist: führe '**Dummy**'-**Schlupfvariable** ein

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

- Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist: multipliziere Zeile mit -1
- Füge Schlupf-/Überschussvariablen ein, um Gleichheitsform (mit positiver rechter Seite) zu bekommen.
- Wo noch keine *Schlupf*variable ist: führe '**Dummy**'-**Schlupfvariable** ein
- Zielfunktion: Alle Dummy-Schlupfvariablen erhalten Koeffizient -1 in der (Maximierungs-)Zielfunktion (und alle anderen Variablen erhalten Koeffizient 0)

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

- Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist: multipliziere Zeile mit -1
- Füge Schlupf-/Überschussvariablen ein, um Gleichheitsform (mit positiver rechter Seite) zu bekommen.
- Wo noch keine *Schlupf*variable ist: führe '**Dummy**'-**Schlupfvariable** ein
- Zielfunktion: Alle Dummy-Schlupfvariablen erhalten Koeffizient -1 in der (Maximierungs-)Zielfunktion (und alle anderen Variablen erhalten Koeffizient 0)

Löse das entstandene Hilfsproblem (startend mit Basislösung $\{\tilde{y}_i = b_i\}_{i=1,\dots,m}$).

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

- Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist: multipliziere Zeile mit -1
- Füge Schlupf-/Überschussvariablen ein, um Gleichheitsform (mit positiver rechter Seite) zu bekommen.
- Wo noch keine *Schlupf*variable ist: führe '**Dummy**'-**Schlupfvariable** ein
- Zielfunktion: Alle Dummy-Schlupfvariablen erhalten Koeffizient -1 in der (Maximierungs-)Zielfunktion (und alle anderen Variablen erhalten Koeffizient 0)

Löse das entstandene Hilfsproblem (startend mit Basislösung $\{\tilde{y}_i = b_i\}_{i=1,\dots,m}$).

Jetzt gilt: Wenn eine optimale Basislösung des Hilfsproblems mit ZFW 0 gefunden wird, dann ist diese eine zulässige Basislösung des Ursprungsproblems. → Starte damit den normalen Simplex ('Phase 2')
Gibt es keine Optimallösung mit ZFW 0 , dann ist das ursprüngliche Problem unzulässig.

Finden einer zulässigen Basislösung

ansonsten: **2-Phasen-Simplex**

Erzeuge Hilfsproblem:

- Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist: multipliziere Zeile mit -1
- Füge Schlupf-/Überschussvariablen ein, um Gleichheitsform (mit positiver rechter Seite) zu bekommen.
- Wo noch keine *Schlupf*variable ist: führe '**Dummy**'-**Schlupfvariable** ein
- Zielfunktion: Alle Dummy-Schlupfvariablen erhalten Koeffizient -1 in der (Maximierungs-)Zielfunktion (und alle anderen Variablen erhalten Koeffizient 0)

Löse das entstandene Hilfsproblem (startend mit Basislösung $\{\tilde{y}_i = b_i\}_{i=1,\dots,m}$).

Jetzt gilt: Wenn eine optimale Basislösung des Hilfsproblems mit ZFW 0 gefunden wird, dann ist diese eine zulässige Basislösung des Ursprungsproblems. → Starte damit den normalen Simplex ('Phase 2')

Gibt es keine Optimallösung mit ZFW 0 , dann ist das ursprüngliche Problem unzulässig. → **Übungsblatt 10**

Beispiel - 2-Phasen-Simplex

$$\max \quad 2x_1 + x_2$$

$$\text{so dass } x_1 + x_2 \geq 1$$

$$-3x_1 + x_2 \geq -10$$

$$x_2 \leq 2$$

$$x_1 \geq 1$$

$$x_1, x_2 \geq 0$$

Beispiel - 2-Phasen-Simplex

$$\begin{aligned} &\max \quad 2x_1 + x_2 \\ \text{so dass} \quad &x_1 + x_2 \geq 1 \\ &-3x_1 + x_2 \geq -10 \\ &x_2 \leq 2 \\ &x_1 \geq 1 \\ &x_1, x_2 \geq 0 \end{aligned}$$

Erstelle Hilfsproblem



Beispiel - 2-Phasen-Simplex

$$\begin{aligned}
 &\max \quad 2x_1 + x_2 \\
 &\text{so dass } x_1 + x_2 \geq 1 \\
 &\quad -3x_1 + x_2 \geq -10 \\
 &\quad x_2 \leq 2 \\
 &\quad x_1 \geq 1 \\
 &\quad x_1, x_2 \geq 0
 \end{aligned}$$

Erstelle Hilfsproblem

Falls ein b_i (im ursprünglichen LP in Normalform) negativ ist: multipliziere Zeile mit -1

$$\begin{aligned}
 &\text{so dass } x_1 + x_2 \geq 1 \\
 &\quad 3x_1 - x_2 \leq 10 \\
 &\quad x_2 \leq 2 \\
 &\quad x_1 \geq 1 \\
 &\quad x_1, x_2 \geq 0
 \end{aligned}$$

Beispiel - 2-Phasen-Simplex

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ \text{so dass} \quad & x_1 + x_2 \geq 1 \\ & -3x_1 + x_2 \geq -10 \\ & x_2 \leq 2 \\ & x_1 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Erstelle Hilfsproblem

Füge Schlupf-
/Überschussvariablen
ein, um Gleichheitsform
(mit positiver rechter
Seite) zu bekommen.

$$\begin{aligned} \text{so dass} \quad & x_1 + x_2 - y_1 = 1 \\ & 3x_1 - x_2 + y_2 = 10 \\ & x_2 + y_3 = 2 \\ & x_1 - y_4 = 1 \\ & x_1, x_2 \geq 0 \\ & y_1, y_2, y_3, y_4 \geq 0 \end{aligned}$$

Beispiel - 2-Phasen-Simplex

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ \text{so dass} \quad & x_1 + x_2 \geq 1 \\ & -3x_1 + x_2 \geq -10 \\ & x_2 \leq 2 \\ & x_1 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Erstelle Hilfsproblem

Wo noch keine
*Schlupf*variable ist:
führe '**Dummy**'-
Schlupfvariable ein

$$\begin{aligned} \text{so dass} \quad & x_1 + x_2 - y_1 + \tilde{y}_1 = 1 \\ & 3x_1 - x_2 + y_2 = 10 \\ & x_2 + y_3 = 2 \\ & x_1 - y_4 + \tilde{y}_4 = 1 \\ & x_1, x_2 \geq 0 \\ & y_1, y_2, y_3, y_4 \geq 0 \\ & \tilde{y}_1, \tilde{y}_4 \geq 0 \end{aligned}$$

Beispiel - 2-Phasen-Simplex

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ \text{so dass} \quad & x_1 + x_2 \geq 1 \\ & -3x_1 + x_2 \geq -10 \\ & x_2 \leq 2 \\ & x_1 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Erstelle Hilfsproblem

Zielfunktion: Alle
Dummy-Schlupfvariablen
erhalten Koeffizient -1 in
der (Maximierungs-
)Zielfunktion (und alle
anderen Variablen
erhalten Koeffizient 0)

$$\begin{aligned} \max \quad & -\tilde{y}_1 - \tilde{y}_4 \\ \text{so dass} \quad & x_1 + x_2 - y_1 + \tilde{y}_1 = 1 \\ & 3x_1 - x_2 + y_2 = 10 \\ & x_2 + y_3 = 2 \\ & x_1 - y_4 + \tilde{y}_4 = 1 \\ & x_1, x_2 \geq 0 \\ & y_1, y_2, y_3, y_4 \geq 0 \\ & \tilde{y}_1, \tilde{y}_4 \geq 0 \end{aligned}$$

Beispiel - 2-Phasen-Simplex

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ \text{so dass} \quad & x_1 + x_2 \geq 1 \\ & -3x_1 + x_2 \geq -10 \\ & x_2 \leq 2 \\ & x_1 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Erstelle Hilfsproblem

$$\begin{aligned} \max \quad & -\tilde{y}_1 - \tilde{y}_4 \\ \text{so dass} \quad & x_1 + x_2 - y_1 + \tilde{y}_1 = 1 \\ & 3x_1 - x_2 + y_2 = 10 \\ & x_2 + y_3 = 2 \\ & x_1 - y_4 + \tilde{y}_4 = 1 \\ & x_1, x_2 \geq 0 \\ & y_1, y_2, y_3, y_4 \geq 0 \\ & \tilde{y}_1, \tilde{y}_4 \geq 0 \end{aligned}$$

Simplextableau

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	0	-1	-1	0
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Beispiel - 2-Phasen-Simplex

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ \text{so dass} \quad & x_1 + x_2 \geq 1 \\ & -3x_1 + x_2 \geq -10 \\ & x_2 \leq 2 \\ & x_1 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Erstelle Hilfsproblem

$$\begin{aligned} \max \quad & -\tilde{y}_1 - \tilde{y}_4 \\ \text{so dass} \quad & x_1 + x_2 - y_1 + \tilde{y}_1 = 1 \\ & 3x_1 - x_2 + y_2 = 10 \\ & x_2 + y_3 = 2 \\ & x_1 - y_4 + \tilde{y}_4 = 1 \\ & x_1, x_2 \geq 0 \\ & y_1, y_2, y_3, y_4 \geq 0 \\ & \tilde{y}_1, \tilde{y}_4 \geq 0 \end{aligned}$$

Simplextableau

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	0	-1	-1	0
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Achtung: Zielfunktion
ist noch nicht in
Basisdarstellung für
Basis $\{y_2, y_3, \tilde{y}_1, \tilde{y}_4\}$

Beispiel - 2-Phasen-Simplex

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ \text{so dass} \quad & x_1 + x_2 \geq 1 \\ & -3x_1 + x_2 \geq -10 \\ & x_2 \leq 2 \\ & x_1 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Erstelle Hilfsproblem

$$\begin{aligned} \max \quad & -\tilde{y}_1 - \tilde{y}_4 \\ \text{so dass} \quad & x_1 + x_2 - y_1 + \tilde{y}_1 = 1 \\ & 3x_1 - x_2 + y_2 = 10 \\ & x_2 + y_3 = 2 \\ & x_1 - y_4 + \tilde{y}_4 = 1 \\ & x_1, x_2 \geq 0 \\ & y_1, y_2, y_3, y_4 \geq 0 \\ & \tilde{y}_1, \tilde{y}_4 \geq 0 \end{aligned}$$

Simplextableau

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	0	-1	-1	0
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Achtung: Zielfunktion
ist noch nicht in
Basisdarstellung für
Basis $\{y_2, y_3, \tilde{y}_1, \tilde{y}_4\}$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
2	1	-1	0	0	-1	0	0	1.5
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Simplextableau

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	0	-1	-1	0
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Achtung: Zielfunktion
ist noch nicht in
Basisdarstellung für
Basis $\{y_2, y_3, \tilde{y}_1, \tilde{y}_4\}$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
2	1	-1	0	0	-1	0	0	1.5
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Jetzt haben wir auch
Zielfunktion in
Basisdarstellung für
Basis $\{y_2, y_3, \tilde{y}_1, \tilde{y}_4\}$

Simplextableau

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	0	-1	-1	0
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Achtung: Zielfunktion
ist noch nicht in
Basisdarstellung für
Basis $\{y_2, y_3, \tilde{y}_1, \tilde{y}_4\}$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
2	1	-1	0	0	-1	0	0	1.5
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Simplextableau

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	0	-1	-1	0
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Achtung: Zielfunktion
ist noch nicht in
Basisdarstellung für
Basis $\{y_2, y_3, \tilde{y}_1, \tilde{y}_4\}$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
2	1	-1	0	0	-1	0	0	1.5
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

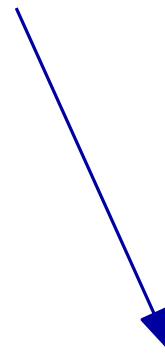
Simplextableau

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	0	-1	-1	0
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Achtung: Zielfunktion
ist noch nicht in
Basisdarstellung für
Basis $\{y_2, y_3, \tilde{y}_1, \tilde{y}_4\}$

Beispiel - 2-Phasen-Simplex

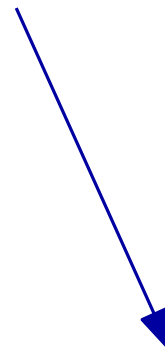
x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
2	1	-1	0	0	-1	0	0	1.5
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5



x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	1	-1	0	0	-3	0	-2	0.5
0	1	-1	0	0	1	1	-1	8.5
0	-1	0	1	0	3	0	-3	9
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
2	1	-1	0	0	-1	0	0	1.5
1	1	-1	0	0	0	1	0	1
3	-1	0	1	0	0	0	0	10
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5



x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	1	-1	0	0	-3	0	-2	0.5
0	1	-1	0	0	1	1	-1	8.5
0	-1	0	1	0	3	0	-3	9
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	-4	-1	-1	0
0	1	-1	0	0	1	1	-1	0.5
0	0	-1	1	0	4	1	-4	9
0	0	1	0	1	-1	-1	1	1.5
1	0	0	0	0	-1	0	1	0.5



x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	1	-1	0	0	-3	0	-2	0.5
0	1	-1	0	0	1	1	-1	8.5
0	-1	0	1	0	3	0	-3	9
0	1	0	0	1	0	0	0	2
1	0	0	0	0	-1	0	1	0.5

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	-4	-1	-1	0
0	1	-1	0	0	1	1	-1	0.5
0	0	-1	1	0	4	1	-4	9
0	0	1	0	1	-1	-1	1	1.5
1	0	0	0	0	-1	0	1	0.5

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	-4	-1	-1	0
0	1	-1	0	0	1	1	-1	0.5
0	0	-1	1	0	4	1	-4	9
0	0	1	0	1	-1	-1	1	1.5
1	0	0	0	0	-1	0	1	0.5

Optimale Lösung für Hilfsproblem:

$$x_1 = 0.5, x_2 = 0.5, y_2 = 9, y_3 = 1.5$$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	-4	-1	-1	0
0	1	-1	0	0	1	1	-1	0.5
0	0	-1	1	0	4	1	-4	9
0	0	1	0	1	-1	-1	1	1.5
1	0	0	0	0	-1	0	1	0.5

Optimale Lösung für Hilfsproblem:

$$x_1 = 0.5, x_2 = 0.5, y_2 = 9, y_3 = 1.5$$

Insbesondere: $\tilde{y}_1, \tilde{y}_4 = 0$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	-4	-1	-1	0
0	1	-1	0	0	1	1	-1	0.5
0	0	-1	1	0	4	1	-4	9
0	0	1	0	1	-1	-1	1	1.5
1	0	0	0	0	-1	0	1	0.5

Optimale Lösung für Hilfsproblem:

$$x_1 = 0.5, x_2 = 0.5, y_2 = 9, y_3 = 1.5$$

Insbesondere: $\tilde{y}_1, \tilde{y}_4 = 0$

→ zulässige Lösung für ursprüngliches Problem

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	-4	-1	-1	0
0	1	-1	0	0	1	1	-1	0.5
0	0	-1	1	0	4	1	-4	9
0	0	1	0	1	-1	-1	1	1.5
1	0	0	0	0	-1	0	1	0.5

Optimale Lösung für Hilfsproblem:

$$x_1 = 0.5, x_2 = 0.5, y_2 = 9, y_3 = 1.5$$

Insbesondere: $\tilde{y}_1, \tilde{y}_4 = 0$

→ zulässige Lösung für ursprüngliches Problem

Phase 2:

x_1	x_2	y_1	y_2	y_3	y_4	$-ZF$
2	1	0	0	0	0	0
0	1	-1	0	0	1	0.5
0	0	-1	1	0	4	9
0	0	1	0	1	-1	1.5
1	0	0	0	0	-1	0.5

ursprüngliche Zielfunktion

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	\tilde{y}_1	\tilde{y}_4	$-ZF$
0	0	0	0	0	-4	-1	-1	0
0	1	-1	0	0	1	1	-1	0.5
0	0	-1	1	0	4	1	-4	9
0	0	1	0	1	-1	-1	1	1.5
1	0	0	0	0	-1	0	1	0.5

Optimale Lösung für Hilfsproblem:

$$x_1 = 0.5, x_2 = 0.5, y_2 = 9, y_3 = 1.5$$

Insbesondere: $\tilde{y}_1, \tilde{y}_4 = 0$

→ zulässige Lösung für ursprüngliches Problem

Phase 2:

x_1	x_2	y_1	y_2	y_3	y_4	$-ZF$
2	1	0	0	0	0	0
0	1	-1	0	0	1	0.5
0	0	-1	1	0	4	9
0	0	1	0	1	-1	1.5
1	0	0	0	0	-1	0.5

ursprüngliche Zielfunktion

Achtung: ZF noch nicht in
Basisdarstellung für Basis
 $\{x_1, x_2, y_2, y_3\}$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	$-ZF$
0	0	1	0	0	-2	1.5
0	1	-1	0	0	1	0.5
0	0	-1	1	0	4	9
0	0	1	0	1	-1	1.5
1	0	0	0	0	-1	0.5

Phase 2:

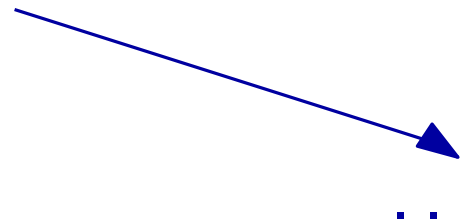
x_1	x_2	y_1	y_2	y_3	y_4	$-ZF$
2	1	0	0	0	0	0
0	1	-1	0	0	1	0.5
0	0	-1	1	0	4	9
0	0	1	0	1	-1	1.5
1	0	0	0	0	-1	0.5

ursprüngliche Zielfunktion

Achtung: ZF noch nicht in
Basisdarstellung für Basis
 $\{x_1, x_2, y_2, y_3\}$

Beispiel - 2-Phasen-Simplex

x_1	x_2	y_1	y_2	y_3	y_4	$-ZF$
0	0	1	0	0	-2	1.5
0	1	-1	0	0	1	0.5
0	0	-1	1	0	4	9
0	0	1	0	1	-1	1.5
1	0	0	0	0	-1	0.5



Phase 2:

x_1	x_2	y_1	y_2	y_3	y_4	$-ZF$
2	1	0	0	0	0	0
0	1	-1	0	0	1	0.5
0	0	-1	1	0	4	9
0	0	1	0	1	-1	1.5
1	0	0	0	0	-1	0.5

ursprüngliche Zielfunktion

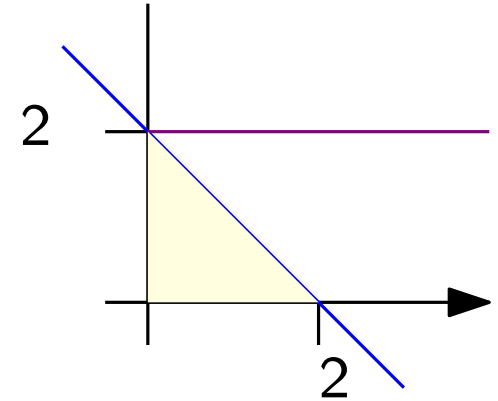
Achtung: ZF noch nicht in
Basisdarstellung für Basis
 $\{x_1, x_2, y_2, y_3\}$

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Konvergenz & Laufzeit des Simplexverfahrens

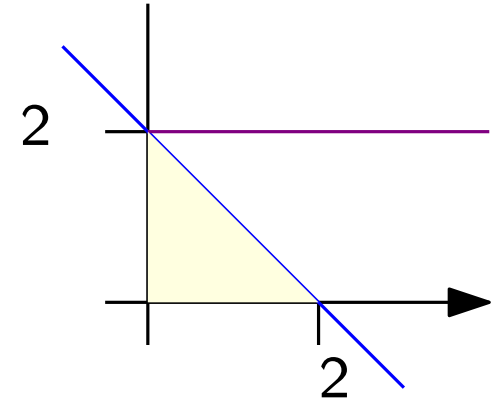
Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?



Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\begin{aligned} \max \quad & x_1 + 2x_2 \\ \text{so dass} \quad & x_1 + x_2 \leq 2 \\ & x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

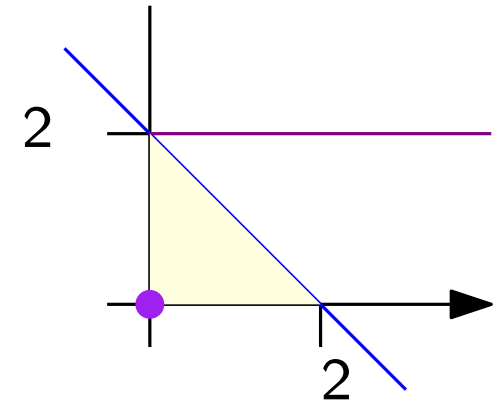
$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2



Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

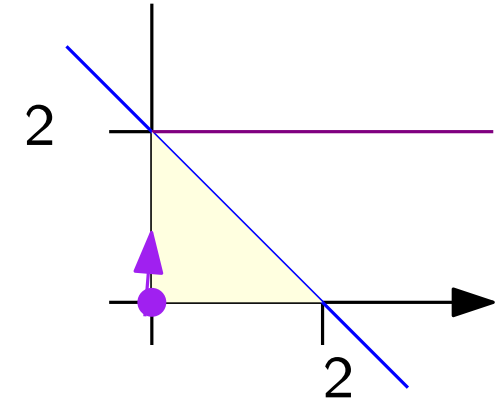
$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2



Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

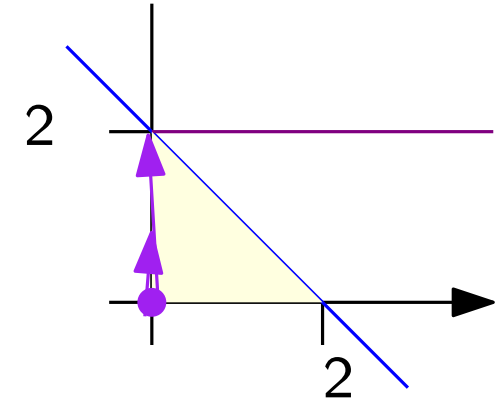
$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2



Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

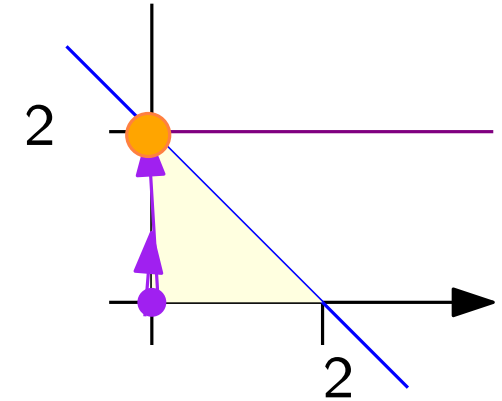
$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2



1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

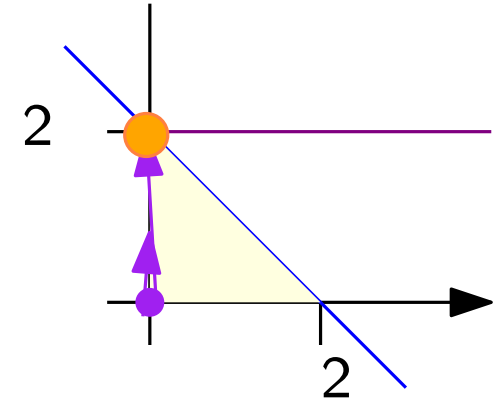
Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$



1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

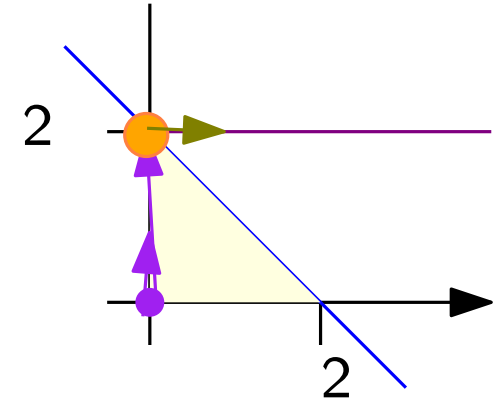
Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$



1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

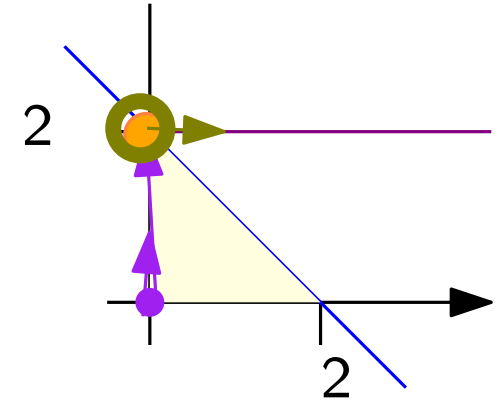
Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$



1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

0	-1	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

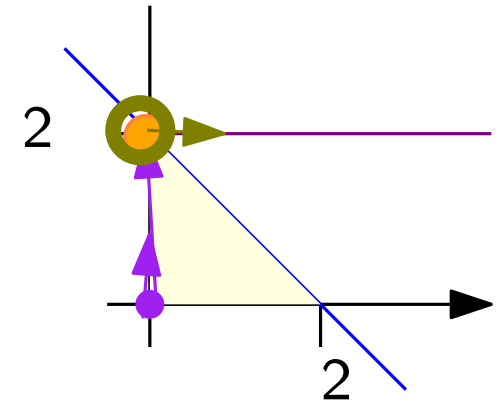
Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$



1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

Zulässige Basislösung heißt **degeneriert**, wenn mindestens eine Basisvariable den Wert 0 hat.

1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

0	-1	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

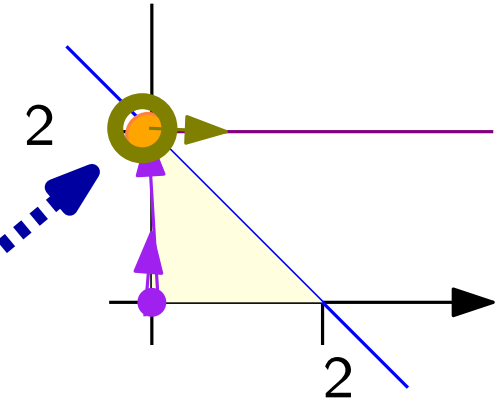
$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

Zulässige Basislösung heißt **degeneriert**, wenn mindestens eine Basisvariable den Wert 0 hat.



1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

0	-1	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

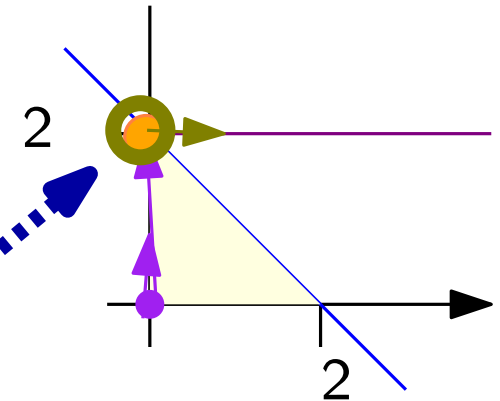
$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

Zulässige Basislösung heißt **degeneriert**, wenn mindestens eine Basisvariable den Wert 0 hat.

Ein LP heißt **degeneriert**, wenn es mindestens eine degenerierte Basislösung hat.



1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

0	-1	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

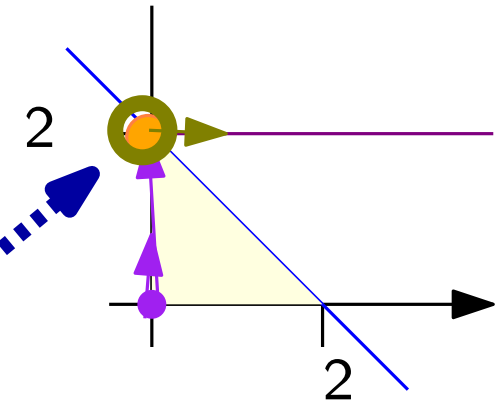
$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

Zulässige Basislösung heißt **degeneriert**, wenn mindestens eine Basisvariable den Wert 0 hat.

Ein LP heißt **degeneriert**, wenn es mindestens eine degenerierte Basislösung hat.

In einem nicht-degenerierten LP wird in jedem Schritt eine bessere Basislösung gefunden.



1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

0	-1	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

$$\max x_1 + 2x_2$$

$$\text{so dass } x_1 + x_2 \leq 2$$

$$x_2 \leq 2$$

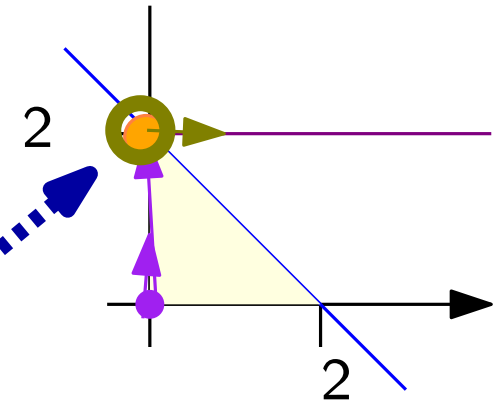
$$x_1, x_2 \geq 0$$

1	2	0	0		0
1	1	1	0		2
0	1	0	1		2

Zulässige Basislösung heißt **degeneriert**, wenn mindestens eine Basisvariable den Wert 0 hat.

Ein LP heißt **degeneriert**, wenn es mindestens eine degenerierte Basislösung hat.

In einem degenerierten LP ist das nicht der Fall, hier ist nur garantiert, dass die neue Lösung nicht schlechter ist.



1	0	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

0	-1	0	-1		-4
1	0	1	-1		0
0	1	0	1		2

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Ist das Simplexverfahren endlich?

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Ist das Simplexverfahren endlich?

Für nicht-degenerierte LP's: ja.

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Ist das Simplexverfahren endlich?

Für nicht-degenerierte LP's: ja.

Für degenerierte LP's kann es zum 'Kreisen' kommen, wenn die Pivotregel nicht gut gewählt ist → nutze **Bland's Pivotregel**
(siehe zum Beispiel Burkard-Zimmermann)

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Ist das Simplexverfahren endlich?

Für nicht-degenerierte LP's: ja.

Für degenerierte LP's kann es zum 'Kreisen' kommen, wenn die Pivotregel nicht gut gewählt ist → nutze **Bland's Pivotregel**
(siehe zum Beispiel Burkard-Zimmermann)

Wie viele Schritte braucht das Simplexverfahren?

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Ist das Simplexverfahren endlich?

Für nicht-degenerierte LP's: ja.

Für degenerierte LP's kann es zum 'Kreisen' kommen, wenn die Pivotregel nicht gut gewählt ist → nutze **Bland's Pivotregel**
(siehe zum Beispiel Burkard-Zimmermann)

Wie viele Schritte braucht das Simplexverfahren?

Man kann Beispiele (bzw. eine Sequenz von Beispielen) generieren, in denen tatsächlich exponentiell viele Basislösungen betrachtet werden → Klee-Minty-Würfel, siehe z.B. Burkard-Zimmermann

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Ist das Simplexverfahren endlich?

Für nicht-degenerierte LP's: ja.

Für degenerierte LP's kann es zum 'Kreisen' kommen, wenn die Pivotregel nicht gut gewählt ist → nutze **Bland's Pivotregel**
(siehe zum Beispiel Burkard-Zimmermann)

Wie viele Schritte braucht das Simplexverfahren?

Man kann Beispiele (bzw. eine Sequenz von Beispielen) generieren, in denen tatsächlich exponentiell viele Basislösungen betrachtet werden → Klee-Minty-Würfel, siehe z.B. Burkard-Zimmermann

Im 'average case' ist das Simplexverfahren schnell!

Konvergenz & Laufzeit des Simplexverfahrens

Wird in jedem Schritt des Simplexverfahrens eine *bessere* Basislösung gefunden?

Falls das LP nicht-degeneriert ist: ja.

Ansonsten ist die neue Lösung zumindest nicht schlechter.

Ist das Simplexverfahren endlich?

Für nicht-degenerierte LP's: ja.

Für degenerierte LP's kann es zum 'Kreisen' kommen, wenn die Pivotregel nicht gut gewählt ist → nutze **Bland's Pivotregel**
(siehe zum Beispiel Burkard-Zimmermann)

Wie viele Schritte braucht das Simplexverfahren?

Man kann Beispiele (bzw. eine Sequenz von Beispielen) generieren, in denen tatsächlich exponentiell viele Basislösungen betrachtet werden → Klee-Minty-Würfel, siehe z.B. Burkard-Zimmermann

Im 'average case' ist das Simplexverfahren schnell!

Bedeutet das, dass lineare Programmierung NP-schwer ist?

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

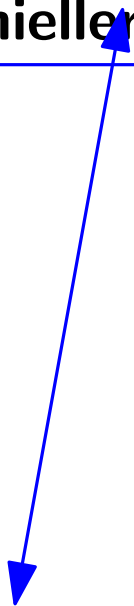
1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's



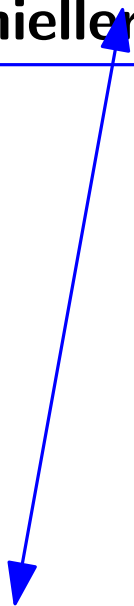
Konsequenz:

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's



Konsequenz:

Lässt sich ein Problem als LP schreiben, dann ist es nicht NP-schwer!

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's
- in der Praxis allerdings nicht zu gebrauchen

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's
- in der Praxis allerdings nicht zu gebrauchen

1984: Karmarkar's Algorithmus: erstes polynomielles
'Innere-Punkte-Verfahren' zum Lösen von LP's (Karmarkar)

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's
- in der Praxis allerdings nicht zu gebrauchen

1984: Karmarkar's Algorithmus: erstes polynomielles
'Innere-Punkte-Verfahren' zum Lösen von LP's (Karmarkar)
- praktisch nutzbar, aber viel langsamer als Simplex

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's
- in der Praxis allerdings nicht zu gebrauchen

1984: Karmarkar's Algorithmus: erstes polynomielles
'Innere-Punkte-Verfahren' zum Lösen von LP's (Karmarkar)
- praktisch nutzbar, aber viel langsamer als Simplex

1990er: Innere-Punkte-Methoden werden konkurrenzfähig, insbesondere für große, dünnbesetzte LP's

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's
- in der Praxis allerdings nicht zu gebrauchen

1984: Karmarkar's Algorithmus: erstes polynomielles
'Innere-Punkte-Verfahren' zum Lösen von LP's (Karmarkar)
- praktisch nutzbar, aber viel langsamer als Simplex

1990er: Innere-Punkte-Methoden werden konkurrenzfähig, insbesondere für große, dünnbesetzte LP's
Auch z.B. für quadratische Programme nutzbar

Komplexität & Lösungsmethoden

1947: Entwicklung Simplexverfahren (Dantzig)

1972: Klee-Minty-Würfel: Simplexverfahren ist im worst-case nicht polynomiell (Klee & Minty)

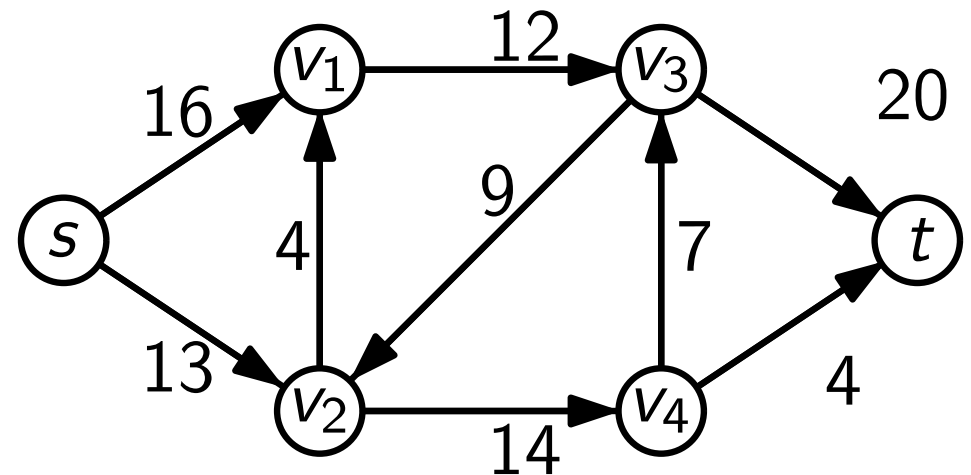
1979: Ellipsoidmethode (Yudin, Nemirovski, Schor, Khachiyan): erster **polynomieller** Algorithmus zum Lösen von LP's
- in der Praxis allerdings nicht zu gebrauchen

1984: Karmarkar's Algorithmus: erstes polynomielles
'Innere-Punkte-Verfahren' zum Lösen von LP's (Karmarkar)
- praktisch nutzbar, aber viel langsamer als Simplex

1990er: Innere-Punkte-Methoden werden konkurrenzfähig, insbesondere für große, dünnbesetzte LP's
Auch z.B. für quadratische Programme nutzbar

Solver wie CPLEX verfügen über Simplexverfahren **und**
Innere-Punkte-Methoden

Flussprobleme als lineare Programme



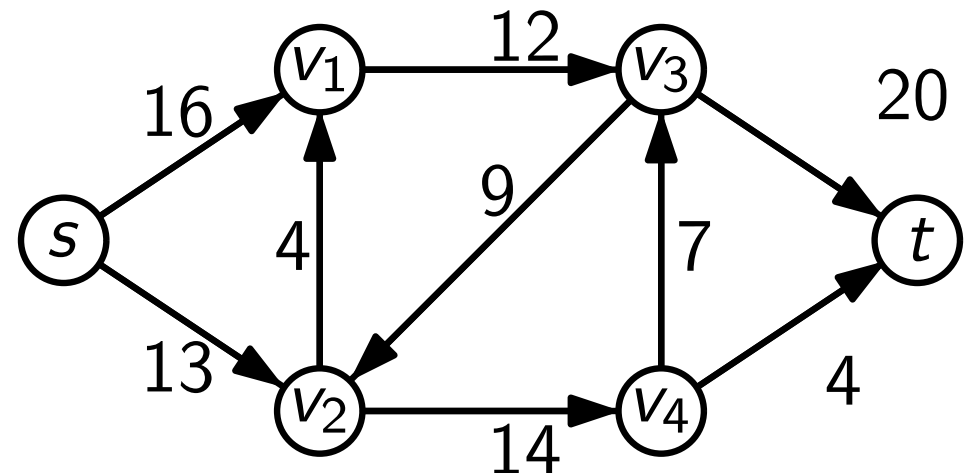
Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:



Flussprobleme als lineare Programme

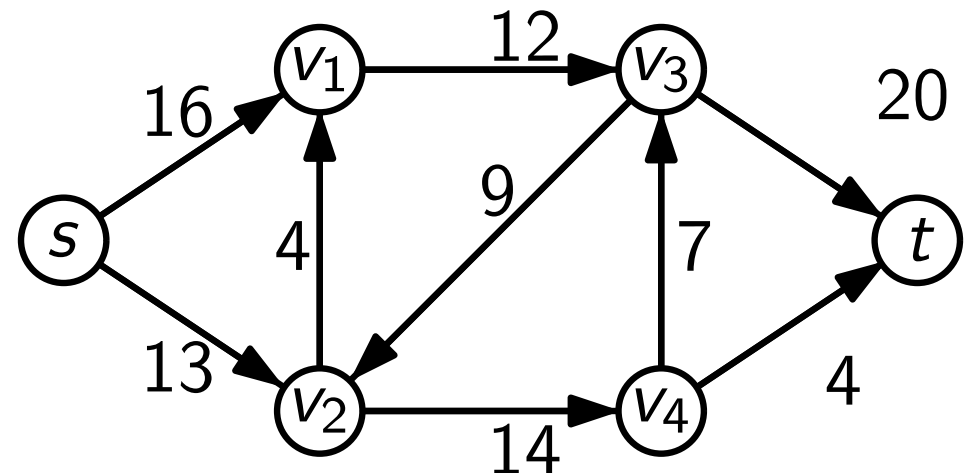
Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen, Nebenbedingungen, Zielfunktion?



Flussprobleme als lineare Programme

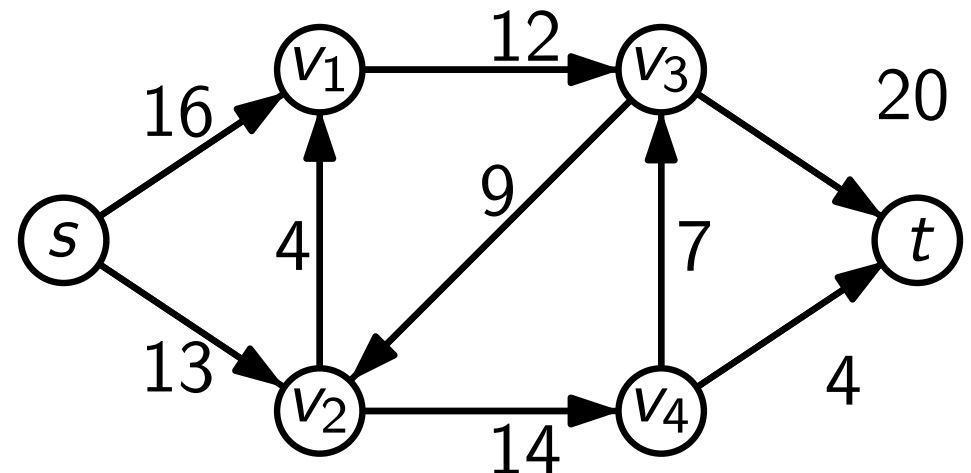
Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e



Flussprobleme als lineare Programme

Maximaler Fluss

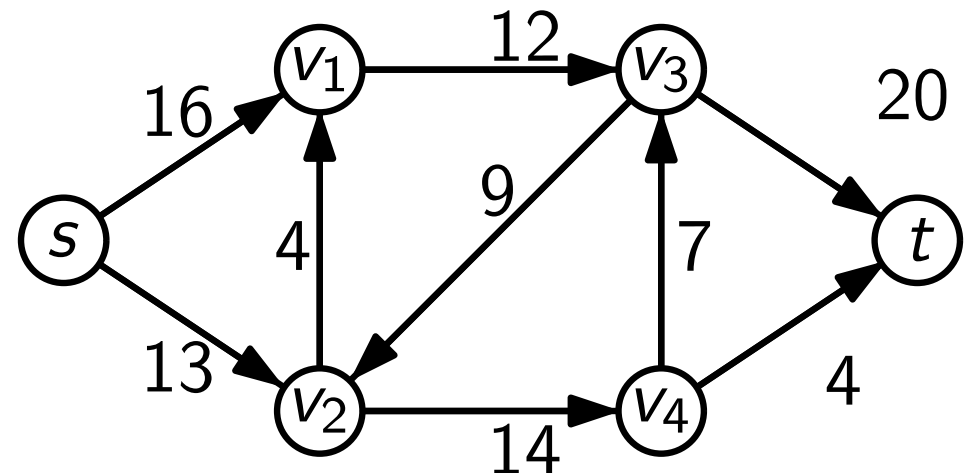
Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$



Flussprobleme als lineare Programme

Maximaler Fluss

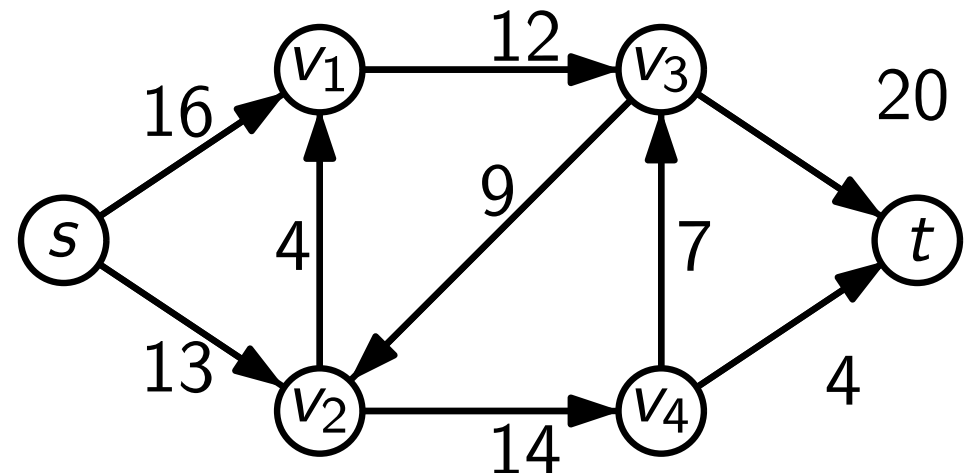
Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$,



Flussprobleme als lineare Programme

Maximaler Fluss

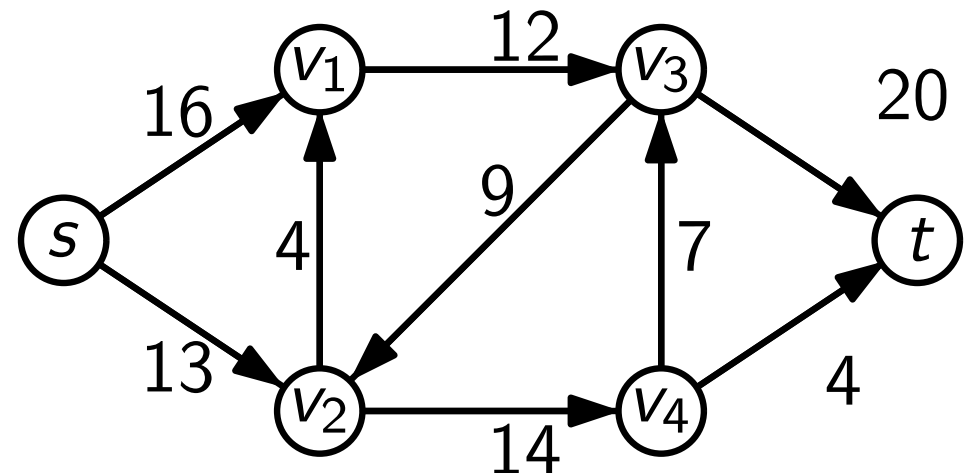
Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \forall e \in E$



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

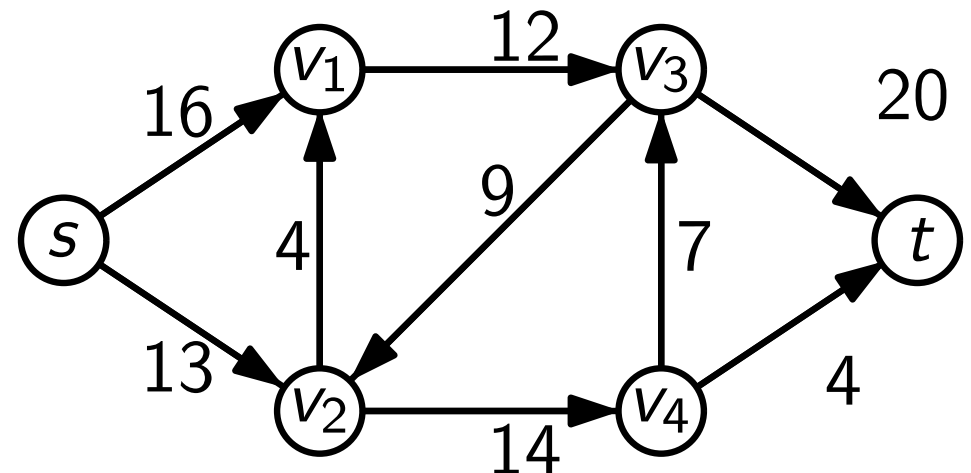
Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

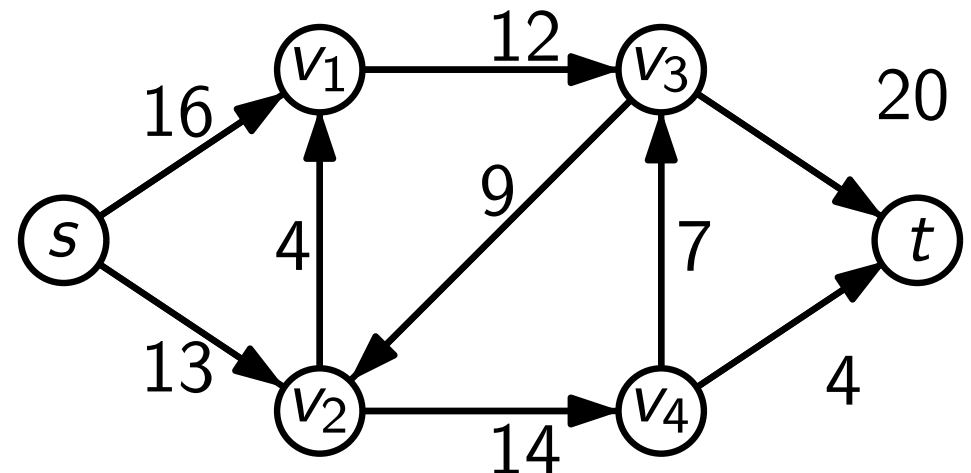
Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \ \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$

LP in Matrixdarstellung



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

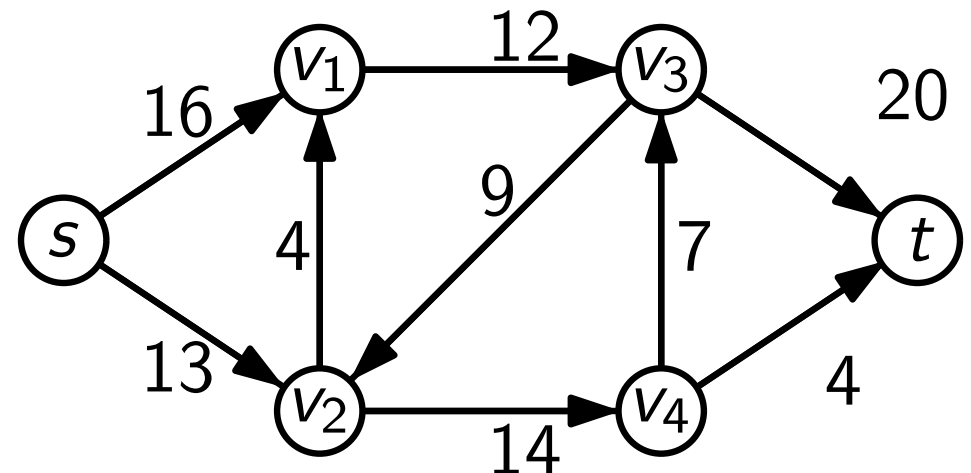
Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \ \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$

LP in Matrixdarstellung

$$\begin{aligned} \max \quad & a_t \cdot f \\ \text{so dass} \quad & A \cdot f = 0 \\ & l \cdot f \leq c \\ & f \geq 0 \end{aligned}$$



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \ \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$

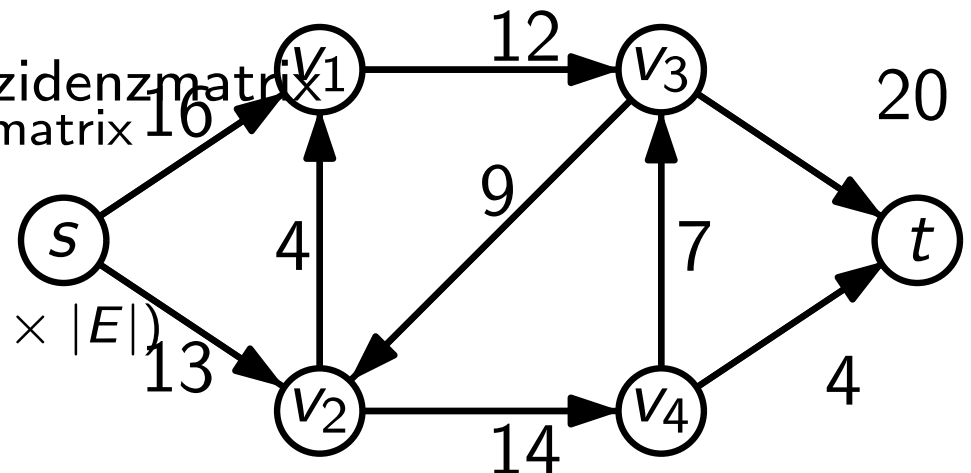
LP in Matrixdarstellung

Knoten- t Zeile von Knoten-Kanten-Inzidenzmatrix
 $\max a_t \cdot f$
 so dass $A \cdot f = 0$
 Knoten-Kanten-Inzidenzmatrix
 von $G[V \setminus \{s, t\}]$

$f \cdot f \leq c$ Einheitsmatrix $(|E| \times |E|)$

$f \geq 0$

komponentenweise



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

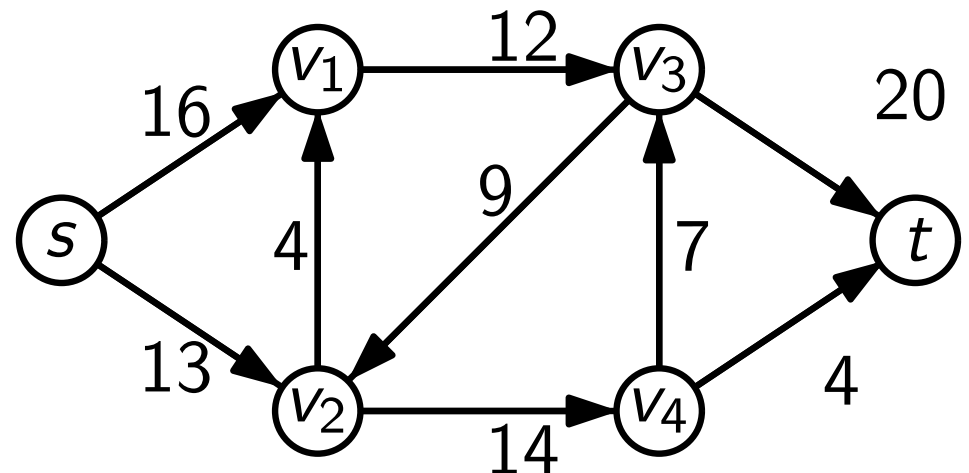
Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$

Überprüfe:

1. Gibt es für jeden Fluss f im Graph eine Zuweisung von Werten f_e , die die Nebenbedingung erfüllt, so dass der Zielfunktionswert

$$\sum_{e \in \delta^-(t)} f_e = |f|?$$



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

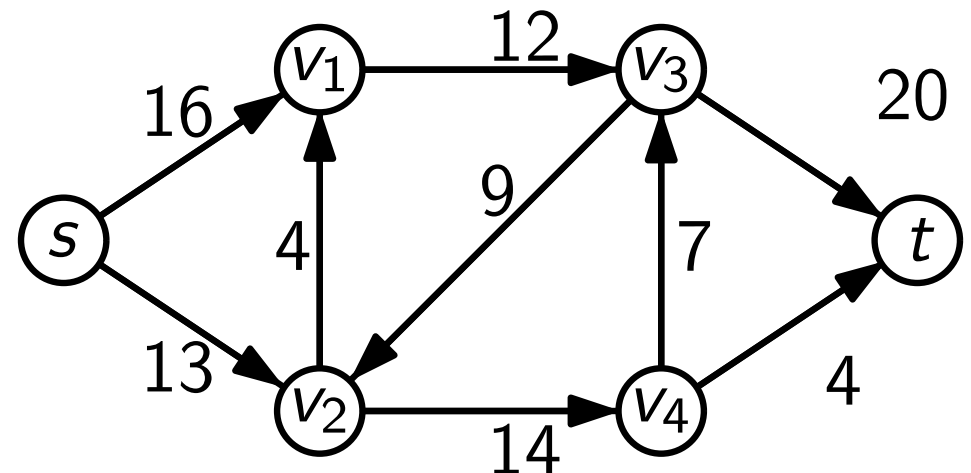
Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$

Überprüfe:

2. Gibt es für jede Zuweisung von Werten f_e , die die Nebenbedingung erfüllt, einen zulässigen Fluss f im Graph, so dass $|f| = \sum_{e \in \delta^-(t)} f_e$?



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

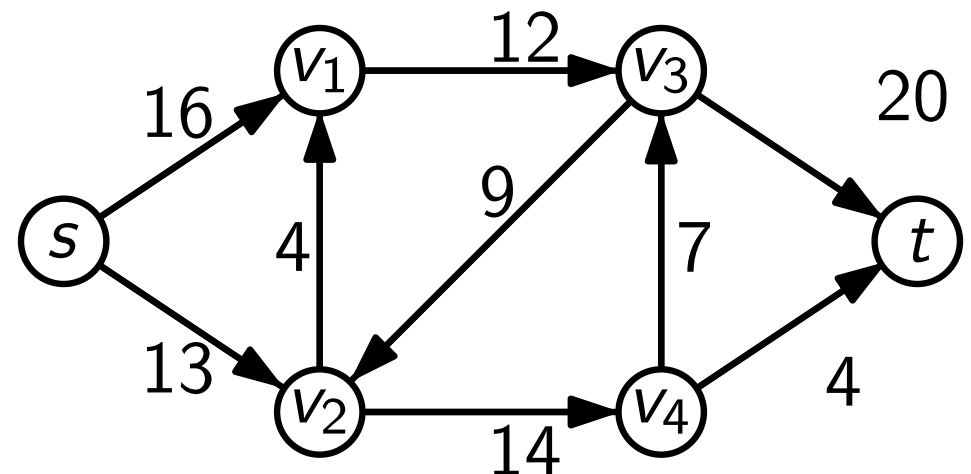
Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$

Überprüfe:

1. Gibt es für jeden Fluss f im Graph eine Zuweisung von Werten f_e , die die Nebenbedingung erfüllt, so dass der Zielfunktionswert

$$\sum_{e \in \delta^-(t)} f_e = |f|?$$



Flussprobleme als lineare Programme

Maximaler Fluss

Gegeben: gerichteter Graph $G = (V, E)$ mit Quelle $s \in V$, Senke $t \in V$ und Kantenkapazitäten $c: E \rightarrow \mathbb{R}_{>0}$.

Gesucht: zulässiger Fluss f , mit maximalem Wert $|f| = \text{Nettozufluss}_f(t)$

Modellierung als lineares (oder ganzzahlig lineares) Programm:

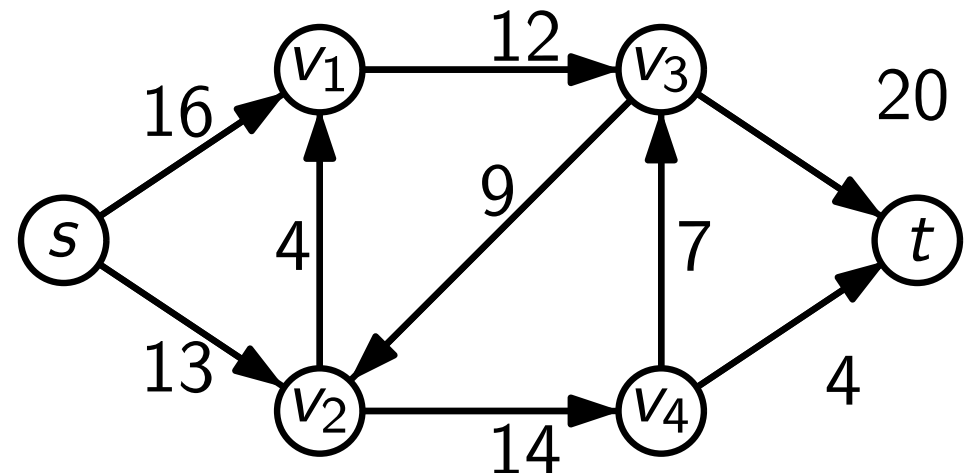
Variablen: f_e für alle $e \in E$: Höhe des Flusses auf Kante e

Nebenbedingungen: $\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0$ für alle $v \in V \setminus \{s, t\}$
 $f_e \leq c_e$ für alle $e \in E$, $f_e \geq 0 \forall e \in E$

Zielfunktion: $\sum_{e \in \delta^-(t)} f_e \rightarrow \max$

Überprüfe:

2. Gibt es für jede Zuweisung von Werten f_e , die die Nebenbedingung erfüllt, einen zulässigen Fluss f im Graph, so dass $|f| = \sum_{e \in \delta^-(t)} f_e$?



Andere Flussprobleme

Schranken

Sei (P) ein Optimierungsproblem mit zulässiger Menge X und Zielfunktion f .

Dann heißt u **obere Schranke** an (P) , falls für alle $x \in X$ $f(x) \leq u$.

u heißt **untere Schranke** an (P) , falls für alle $x \in X$ $f(x) \geq l$.

Obere Schranke für das Produktionsplanungsbeispiel

Eine Firma stellt zwei Produkte P1 und P2 her. Der Verkaufserlös von P1 beträgt 3 Euro, der von P2 beträgt 5 Euro. Wir gehen davon aus, dass alle produzierten Einheiten auch verkauft werden können.

Zur Herstellung der Produkte sind drei Maschinen A, B, C nötig. Die Bearbeitungszeiten der Produkte auf den drei Maschinen und die Kapazität der Maschinen sind wie folgt angegeben:

	Zeit für P1	Zeit für P2	Kapazität der Maschine
Maschine A	1	2	170
Maschine B	1	1	150
Maschine C	-	3	180

Frage: Wie viel soll die Firma von welchem Produkt produzieren, um ihren Gewinn zu maximieren?

Obere Schranke für das Produktionsplanungsbeispiel

Eine Firma stellt zwei Produkte P1 und P2 her. Der Verkaufserlös von P1 beträgt 3 Euro, der von P2 beträgt 5 Euro. Wir gehen davon aus, dass alle produzierten Einheiten auch verkauft werden können.

Zur Herstellung der Produkte sind drei Maschinen A, B, C nötig. Die Bearbeitungszeiten der Produkte auf den drei Maschinen und die Kapazität der Maschinen sind wie folgt angegeben:

	Zeit für P1	Zeit für P2	Kapazität der Maschine
Maschine A	1	2	170
Maschine B	1	1	150
Maschine C	-	3	180

Frage: Wie viel soll die Firma von welchem Produkt produzieren, um ihren Gewinn zu maximieren?

Mathematisches Programm

$$\begin{array}{ll}
 \max & 3x_1 + 5x_2 \\
 \text{so dass} & x_1 + 2x_2 \leq 170 \\
 & x_1 + x_2 \leq 150 \\
 & 3x_2 \leq 180 \\
 & x_1, x_2 \geq 0
 \end{array}$$

x_1 : Anzahl Produkte P1
 x_2 : Anzahl Produkte P2

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)	$\max 3x_1 + 5x_2$	
	so dass $x_1 + 2x_2$	≤ 170
	$x_1 + x_2$	≤ 150
	$3x_2$	≤ 180
	x_1, x_2	≥ 0
x_1 : Anzahl Produkte P1		
x_2 : Anzahl Produkte P2		

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)	\max	$3x_1 + 5x_2$	
	so dass	$x_1 + 2x_2$	≤ 170
	x_1 : Anzahl Produkte P1	$x_1 + x_2$	≤ 150
	x_2 : Anzahl Produkte P2	$3x_2$	≤ 180
		x_1, x_2	≥ 0

Kann ich leicht obere Schranken für den Erlös ablesen?

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)	$\max 3x_1 + 5x_2$	
	so dass $x_1 + 2x_2$	≤ 170
	$x_1 + x_2$	≤ 150
	$3x_2$	≤ 180
	x_1, x_2	≥ 0

Kann ich leicht obere Schranken für den Erlös ablesen?

510 \leftarrow obere Schranke an den Erlös

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

$$\max 3x_1 + 5x_2$$

$$\text{so dass } x_1 + 2x_2 \leq 170$$

x_1 : Anzahl Produkte P1

$$x_1 + x_2 \leq 150$$

x_2 : Anzahl Produkte P2

$$3x_2 \leq 180$$

$$x_1, x_2 \geq 0$$

Kann ich leicht obere Schranken für den Erlös ablesen?

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2) \leq 3 \cdot 170 = 510 \leftarrow \text{obere Schranke an den Erlös}$$

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{array}{ll} \max & 3x_1 + 5x_2 \\ \text{so dass} & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{array}$$

Kann ich leicht obere Schranken für den Erlös ablesen?

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2) \leq 3 \cdot 170 = 510 \leftarrow \text{obere Schranke an den Erlös}$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2) \leq 5 \cdot 150 = 750 \leftarrow \text{obere Schranke an den Erlös}$$

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{array}{ll} \max & 3x_1 + 5x_2 \\ \text{so dass} & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{array}$$

Kann ich leicht obere Schranken für den Erlös ablesen?

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2) \leq 3 \cdot 170 = 510 \leftarrow \text{obere Schranke an den Erlös}$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2) \leq 5 \cdot 150 = 750$$

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + x_2) + 1 \cdot (3x_2) \leq 3 \cdot 150 + 1 \cdot 180 = 630$$

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{array}{ll} \max & 3x_1 + 5x_2 \\ \text{so dass} & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{array}$$

Kann ich leicht obere Schranken für den Erlös ablesen?

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2) \leq 3 \cdot 170 = 510 \leftarrow \text{obere Schranke an den Erlös}$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2) \leq 5 \cdot 150 = 750$$

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + x_2) + 1 \cdot (3x_2) \leq 3 \cdot 150 + 1 \cdot 180 = 630$$

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{aligned} \max & 3x_1 + 5x_2 \\ \text{so dass} & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Kann ich leicht obere Schranken für den Erlös ablesen?

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2) \leq 3 \cdot 170 = 510 \leftarrow \text{obere Schranke an den Erlös}$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2) \leq 5 \cdot 150 = 750$$

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + x_2) + 1 \cdot (3x_2) \leq 3 \cdot 150 + 1 \cdot 180 = 630$$

$$3x_1 + 5x_2 \leq \lambda_1 \cdot (x_1 + 2x_2) + \lambda_2 \cdot (x_1 + x_2) + \lambda_3 \cdot (3x_2) \leq \lambda_1 \cdot 170 + \lambda_2 \cdot 150 + \lambda_3 \cdot 180$$

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{array}{ll} \max & 3x_1 + 5x_2 \\ \text{so dass} & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{array}$$

Kann ich leicht obere Schranken für den Erlös ablesen?

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2) \leq 3 \cdot 170 = 510 \leftarrow \text{obere Schranke an den Erlös}$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2) \leq 5 \cdot 150 = 750$$

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + x_2) + 1 \cdot (3x_2) \leq 3 \cdot 150 + 1 \cdot 180 = 630$$

Suche $\lambda_1, \lambda_2, \lambda_3 \geq 0$

so dass $3 \leq \lambda_1 + \lambda_2$ und $5 \leq 2 \cdot \lambda_1 + \lambda_2 + 3 \cdot \lambda_3$

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{array}{ll} \max & 3x_1 + 5x_2 \\ \text{so dass} & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{array}$$

Kann ich leicht obere Schranken für den Erlös ablesen?

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2) \leq 3 \cdot 170 = 510 \leftarrow \text{obere Schranke an den Erlös}$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2) \leq 5 \cdot 150 = 750$$

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + x_2) + 1 \cdot (3x_2) \leq 3 \cdot 150 + 1 \cdot 180 = 630$$

Suche $\lambda_1, \lambda_2, \lambda_3 \geq 0$

so dass $3 \leq \lambda_1 + \lambda_2$ und $5 \leq 2 \cdot \lambda_1 + \lambda_2 + 3 \cdot \lambda_3$

und $170 \cdot \lambda_1 + 150 \cdot \lambda_2 + 180 \cdot \lambda_3$ minimal

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{aligned} \max \quad & 3x_1 + 5x_2 \\ \text{so dass} \quad & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Kann ich leicht obere

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2)$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2)$$

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2)$$

Das lineare Programm (D)

$$\begin{aligned} \min \quad & 170 \cdot \lambda_1 + 150 \cdot \lambda_2 + 180 \cdot \lambda_3 \\ \text{so dass} \quad & \lambda_1 + \lambda_2 \geq 3 \\ & 2 \cdot \lambda_1 + \lambda_2 + 3 \cdot \lambda_3 \geq 5 \\ & \lambda_1, \lambda_2, \lambda_3 \geq 0 \end{aligned}$$

nennen wir das **duale** Programm zu (P).

Suche $\lambda_1, \lambda_2, \lambda_3 \geq 0$

so dass $3 \leq \lambda_1 + \lambda_2$ und $5 \leq 2 \cdot \lambda_1 + \lambda_2 + 3 \cdot \lambda_3$

und $170 \cdot \lambda_1 + 150 \cdot \lambda_2 + 180 \cdot \lambda_3$ minimal

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)

x_1 : Anzahl Produkte P1

x_2 : Anzahl Produkte P2

$$\begin{aligned} \max \quad & 3x_1 + 5x_2 \\ \text{so dass} \quad & x_1 + 2x_2 \leq 170 \\ & x_1 + x_2 \leq 150 \\ & 3x_2 \leq 180 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Kann ich leicht obere

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2)$$

$$3x_1 + 5x_2 \leq 5 \cdot (x_1 + x_2)$$

$$3x_1 + 5x_2 \leq 3 \cdot (x_1 + 2x_2)$$

Das lineare Programm (D)

$$\begin{aligned} \min \quad & 170 \cdot \lambda_1 + 150 \cdot \lambda_2 + 180 \cdot \lambda_3 \\ \text{so dass} \quad & \lambda_1 + \lambda_2 \geq 3 \\ & 2 \cdot \lambda_1 + \lambda_2 + 3 \cdot \lambda_3 \geq 5 \\ & \lambda_1, \lambda_2, \lambda_3 \geq 0 \end{aligned}$$

nennen wir das **duale** Programm zu (P).

Suche $\lambda_1, \lambda_2, \lambda_3 \geq 0$

so dass $3 \leq \lambda_1 + \lambda_2$ und $5 \leq 2 \cdot \lambda_1 + \lambda_2 + 3 \cdot \lambda_3$

und $170 \cdot \lambda_1 + 150 \cdot \lambda_2 + 180 \cdot \lambda_3$ minimal

Suche: Obere Schranke für den Erlös

Mathematisches Programm (P)	$\max 3x_1 + 5x_2$	
	so dass $x_1 + 2x_2$	≤ 170
x_1 : Anzahl Produkte P1	$x_1 + x_2$	≤ 150
x_2 : Anzahl Produkte P2	$3x_2$	≤ 180
	x_1, x_2	≥ 0

Kann ich leicht obere Schranken für den Erlös ablesen?

Mathematisches Programm: suche *beste* obere Schranke an den Erlös

Das duale Programm

Wir betrachten ein lineares Programm (P):

$$\max \quad c^t \cdot x$$

$$\text{so dass } A \cdot x \leq b$$

$$x \geq 0$$

Das **duale Programm** (D) zu (P) ('Berechne die bestmögliche obere Schranke von (P)) ist gegeben als $\min \quad b^t \cdot \pi$

$$\text{so dass } A^t \cdot \pi \geq c$$

$$\pi \geq 0$$

Das duale Programm

Wir betrachten ein lineares Programm (P):

$$\max \quad c^t \cdot x$$

$$\text{so dass } A \cdot x \leq b$$

$$x \geq 0$$

Das **duale Programm** (D) zu (P) ('Berechne die bestmögliche obere Schranke von (P)) ist gegeben als $\min \quad b^t \cdot \pi$

$$\text{so dass } A^t \cdot \pi \geq c$$

$$\pi \geq 0$$

Es gilt: $c^t \cdot x \leq b^t \cdot \pi$ für alle x aus dem zulässigen Bereich von (P) und π aus dem zulässigen Bereich von (D).

Diese Eigenschaft nennen wir **schwache Dualität**.

Das duale Programm

Wir betrachten ein lineares Programm (P):

$$\max \quad c^t \cdot x$$

$$\text{so dass } A \cdot x \leq b$$

$$x \geq 0$$

Das **duale Programm** (D) zu (P) ('Berechne die bestmögliche obere Schranke von (P)) ist gegeben als $\min \quad b^t \cdot \pi$

$$\text{so dass } A^t \cdot \pi \geq c$$

$$\pi \geq 0$$

Starke Dualität: Seien (P) und (D) zueinander duale lineare Programme. Dann gelten die folgenden Beziehungen:

- 1.) Ist (P) unbeschränkt, dann ist (D) unzulässig.
- 2.) Ist (D) unbeschränkt, dann ist (P) unzulässig.
- 3.) Sei x zulässig für (P) und π zulässig für (D) und gelte $c^t x = b^t \pi$. Dann sind x und π optimal.
- 4.) Hat (P) eine endliche Optimallösung, so auch (D) und die optimalen Zielfunktionswerte von (P) und (D) sind gleich.

Das duale Programm

Für ein allgemeines lineares Programm (P)

$$(P) \max \quad c^t x$$

$$\text{so dass } a_i \cdot x \leq b_i \quad i \in M_1$$

$$a_i \cdot x \geq b_i \quad i \in M_2$$

$$a_i \cdot x = b_i \quad i \in M_3$$

$$x_i \geq 0 \quad i \in N_1$$

$$x_i \leq 0 \quad i \in N_2$$

$$x_i \leq 0 \quad i \in N_3$$

Starke Dualität: Seien (P) und (D) zueinander duale lineare Programme.

Dann gelten die folgenden Beziehungen:

- 1.) Ist (P) unbeschränkt, dann ist (D) unzulässig.
- 2.) Ist (D) unbeschränkt, dann ist (P) unzulässig.
- 3.) Sei x zulässig für (P) und π zulässig für (D) und gelte $c^t x = b^t \pi$.
Dann sind x und π optimal.
- 4.) Hat (P) eine endliche Optimallösung, so auch (D) und die optimalen Zielfunktionswerte von (P) und (D) sind gleich.

Das duale Programm

Für ein allgemeines lineares Programm (P) ist das duale Programm gegeben durch (D) :

$$(P) \max \quad c^t x$$

$$\text{so dass } a_{i \cdot} x \leq b_i \quad i \in M_1$$

$$a_{i \cdot} x \geq b_i \quad i \in M_2$$

$$a_{i \cdot} x = b_i \quad i \in M_3$$

$$x_i \geq 0 \quad i \in N_1$$

$$x_i \leq 0 \quad i \in N_2$$

$$x_i \text{ frei} \quad i \in N_3$$

$$(D) \min \quad b^t \pi$$

$$\text{so dass } a_{\cdot j}^t \pi = c_j \quad j \in N_1$$

$$a_{\cdot j}^t \pi \geq c_j \quad j \in N_2$$

$$a_{\cdot j}^t \pi \leq c_j \quad j \in N_3$$

$$\pi_i \geq 0 \quad i \in M_1$$

$$\pi_i \leq 0 \quad i \in M_2$$

$$\pi_i \text{ frei} \quad i \in M_3$$

Starke Dualität: Seien (P) und (D) zueinander duale lineare Programme.

Dann gelten die folgenden Beziehungen:

1.) Ist (P) unbeschränkt, dann ist (D) unzulässig.

2.) Ist (D) unbeschränkt, dann ist (P) unzulässig.

3.) Sei x zulässig für (P) und π zulässig für (D) und gelte $c^t x = b^t \pi$.

Dann sind x und π optimal.

4.) Hat (P) eine endliche Optimallösung, so auch (D) und die optimalen Zielfunktionswerte von (P) und (D) sind gleich.

Das duale Programm

Für ein allgemeines lineares Programm (P) ist das duale Programm gegeben durch (D) :

$$(P) \max \quad c^t x$$

$$\text{so dass } a_{i \cdot} x \leq b_i \quad i \in M_1$$

$$a_{i \cdot} x \geq b_i \quad i \in M_2$$

$$a_{i \cdot} x = b_i \quad i \in M_3$$

$$x_i \geq 0 \quad i \in N_1$$

$$x_i \leq 0 \quad i \in N_2$$

$$x_i \leq 0 \quad i \in N_3$$

$$(D) \min \quad b^t \pi$$

$$\text{so dass } a_{\cdot j}^t \pi = c_j \quad j \in N_1$$

$$a_{\cdot j}^t \pi \geq c_j \quad j \in N_2$$

$$a_{\cdot j}^t \pi \leq c_j \quad j \in N_3$$

$$\pi_i \geq 0 \quad i \in M_1$$

$$\pi_i \leq 0 \quad i \in M_2$$

$$\pi_i \geq 0 \quad i \in M_3$$

Ist (D) dual zu (P) , dann ist auch (P) dual zu (D) .

Dualisieren - Beispiel

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

$$\text{so dass } x_1 + 3x_2 \geq 4$$

$$x_1 + x_2 \leq 0$$

$$x_1 - x_2 + 2x_3 \leq 6$$

Dualisieren - Beispiel

$$\max \quad x_1 - 2x_2 + 2x_3$$

$$\text{so dass } x_1 + 3x_2 \geq 4 \quad \pi_1$$

$$x_1 + x_2 \leq 0 \quad \pi_2$$

$$x_1 - x_2 + 2x_3 \leq 6 \quad \pi_3$$

Dualisieren - Beispiel

$$\max \quad x_1 - 2x_2 + 2x_3$$

$$\text{so dass } x_1 + 3x_2 \geq 4 \quad \pi_1$$

$$x_1 + x_2 \leq 0 \quad \pi_2$$

$$x_1 - x_2 + 2x_3 \leq 6 \quad \pi_3$$

$$\min \quad 4\pi_1 + 6\pi_3$$

$$\text{so dass } \pi_1 + \pi_2 + \pi_3 = 1$$

$$3\pi_1 + \pi_2 - \pi_3 = -2$$

$$2\pi_3 = 2$$

$$\pi_1 \leq 0, \pi_2 \geq 0, \pi_3 \geq 0$$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i :

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i :

$(P) \max \quad c^t x$

so dass $a_i \cdot x \leq b_i \quad i \in M_1$

$a_i \cdot x \geq b_i \quad i \in M_2$

$a_i \cdot x = b_i \quad i \in M_3$

$x_i \geq 0 \quad i \in N_1$

$x_i \leq 0 \quad i \in N_2$

$x_i \leq 0 \quad i \in N_3$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_j)^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i :

$(P) \max \quad c^t x$

so dass $a_i \cdot x \leq b_i \quad i \in M_1 \quad \rightarrow \pi_i \geq 0$

$a_i \cdot x \geq b_i \quad i \in M_2 \quad \rightarrow \pi_i \leq 0$

$a_i \cdot x = b_i \quad i \in M_3 \quad \rightarrow \pi_i \geq 0$

$x_i \geq 0 \quad i \in N_1$

$x_i \geq 0 \quad i \in N_2$

$x_i \leq 0 \quad i \in N_3$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_j)^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i :

$(P) \max \quad c^t x$

so dass $a_i \cdot x \leq b_i \quad i \in M_1 \quad \rightarrow \pi_i \geq 0 \quad u_i := \pi_i(b_i - a_i \cdot x) \geq 0$

$a_i \cdot x \geq b_i \quad i \in M_2 \quad \rightarrow \pi_i \leq 0 \quad u_i := \pi_i(b_i - a_i \cdot x) \geq 0$

$a_i \cdot x = b_i \quad i \in M_3 \quad \rightarrow \pi_i \geq 0 \quad u_i := \pi_i \cdot 0 = 0$

$x_i \geq 0 \quad i \in N_1$

$x_i \geq 0 \quad i \in N_2$

$x_i \leq 0 \quad i \in N_3$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_j)^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i :

$(P) \max \quad c^t x$

so dass $a_i \cdot x \leq b_i \quad i \in M_1 \quad \rightarrow \pi_i \geq 0 \quad u_i := \pi_i(b_i - a_i \cdot x) \geq 0$

$a_i \cdot x \geq b_i \quad i \in M_2 \quad \rightarrow \pi_i \leq 0 \quad u_i := \pi_i(b_i - a_i \cdot x) \geq 0$

$a_i \cdot x = b_i \quad i \in M_3 \quad \rightarrow \pi_i \geq 0 \quad u_i := \pi_i \cdot 0 = 0$

$x_i \geq 0 \quad i \in N_1$

$x_i \geq 0 \quad i \in N_2$

$x_i \leq 0 \quad i \in N_3$

Analog: zeige $v_j \geq 0$ für alle j (2.)

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $u_j \geq 0$ für alle j : ✓

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $u_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $u_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Dann gilt:

$u_i = 0 \quad \forall i$ und $v_j = 0 \quad \forall j$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $u_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Dann gilt:

$u_i = 0 \quad \forall i$ und $v_j = 0 \quad \forall j \Leftrightarrow u = 0$ und $v = 0$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $v_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Dann gilt:

$$\begin{aligned} u_i = 0 \quad \forall i \text{ und } v_j = 0 \quad \forall j &\Leftrightarrow u = 0 \text{ und } v = 0 \\ &\Leftrightarrow \pi^t(b - Ax) = 0 \text{ und } x^t(A^t\pi - c) = 0 \end{aligned}$$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $v_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Dann gilt:

$$\begin{aligned} u_i = 0 \quad \forall i \text{ und } v_j = 0 \quad \forall j &\Leftrightarrow u = 0 \text{ und } v = 0 \\ &\Leftrightarrow \pi^t(b - Ax) = 0 \text{ und } x \cdot (A^t \pi - c) = 0 \\ &\Leftrightarrow \pi^t(b - Ax) + x \cdot (A^t \pi - c) = 0 \end{aligned}$$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $v_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Dann gilt:

$$\begin{aligned}
 u_i = 0 \quad \forall i \text{ und } v_j = 0 \quad \forall j &\Leftrightarrow u = 0 \text{ und } v = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) = 0 \text{ und } x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) + x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^tb - \pi^tAx + x^tA^t\pi - x^tc = 0
 \end{aligned}$$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $v_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Dann gilt:

$$\begin{aligned}
 u_i = 0 \quad \forall i \text{ und } v_j = 0 \quad \forall j &\Leftrightarrow u = 0 \text{ und } v = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) = 0 \text{ und } x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) + x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^tb - \pi^tAx + x^tA^t\pi - x^tc = 0 \\
 &\Leftrightarrow \pi^tb = x^tc
 \end{aligned}$$

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_{\cdot j})^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $v_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

Dann gilt:

$$\begin{aligned}
 u_i = 0 \quad \forall i \text{ und } v_j = 0 \quad \forall j &\Leftrightarrow u = 0 \text{ und } v = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) = 0 \text{ und } x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) + x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^tb - \pi^tAx + x^tA^t\pi - x^tc = 0 \\
 &\Leftrightarrow \pi^tb = x^tc
 \end{aligned}$$

starke Dualität $\longrightarrow \Leftrightarrow x$ und π sind optimal

Satz vom komplementären Schlupf

Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D) .

Definiere $u_i := \pi_i(b_i - a_i \cdot x) \quad \forall i = 1, \dots, m$

und $v_j := x_j((a_j)^t \pi - c_j) \quad \forall j = 1, \dots, n$.

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn $u_i = 0$ für alle $i = 1, \dots, m$ und $v_j = 0$ für alle $j = 1, \dots, n$.

Beweis:

1. Zeige: $u_i \geq 0$ für alle i : ✓ 2. Zeige: $v_j \geq 0$ für alle j : ✓

Sei $u := \sum_i u_i$ und $v := \sum_j v_j$

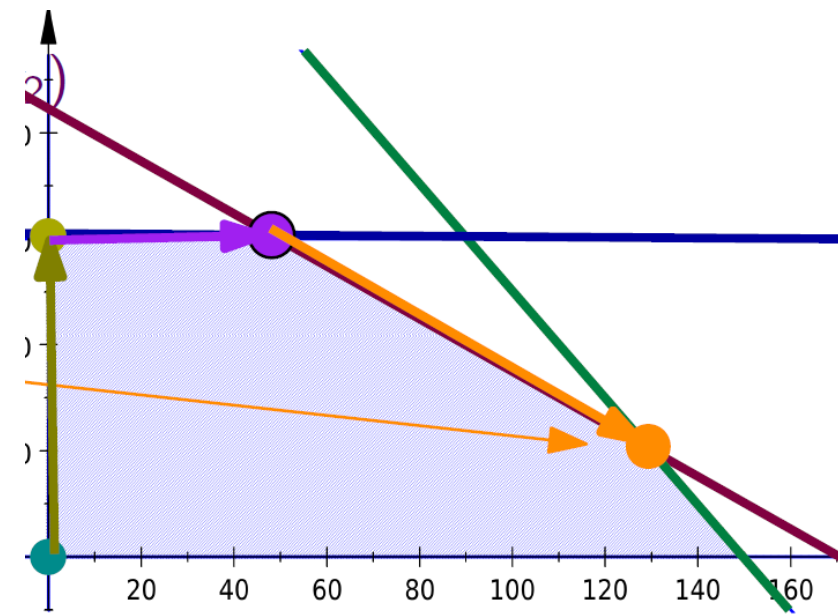
Dann gilt:

$$\begin{aligned}
 u_i = 0 \quad \forall i \text{ und } v_j = 0 \quad \forall j &\Leftrightarrow u = 0 \text{ und } v = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) = 0 \text{ und } x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^t(b - Ax) + x^t(A^t\pi - c) = 0 \\
 &\Leftrightarrow \pi^tb - \pi^tAx + x^tA^t\pi - x^tc = 0 \\
 &\Leftrightarrow \pi^tb = x^tc
 \end{aligned}$$

starke Dualität $\longrightarrow \Leftrightarrow x$ und π sind optimal



Stichworte heute



Mathematische Programmierung:

lineare Programme, Simplexalgorithmus, degeneriert, starke und schwache Dualität, Satz vom komplementären Schlupf, *Flussprobleme als LP*

Verfahren zum Lösen mathematischer Programme:

Simplexverfahren, (Ellipsoid-Methode, Karmarkar's Verfahren, Innere-Punkte-Verfahren)