

Lineares Programmieren

Algorithmentechnik WS 09/10

Dorothea Wagner | 7. Januar 2010

FAKULTÄT FÜR INFORMATIK, INSTITUT FÜR THEORETISCHE INFORMATIK



Beispiel: Einfache Bäckerrechnung



Weizenmischbrot

- Zutaten für eine Kiste Weizenmischbrot:
 - 12kg Weizenmehl
 - 8kg Wasser

Gewinn pro Kiste: 20 Euro



Mehrkornbrot

Zutaten für eine Kiste Mehrkornbrot

- - 6kg Weizenmehl
 - 12kg Wasser
 - 10kg Mischkornschtrot

Gewinn pro Kiste: 60 Euro

■ Der Bäcker möchte viel Geld verdienen!

Beispiel: Einfache Bäckerrechnung



Weizenmischbrot

- Zutaten für eine Kiste Weizenmischbrot:
 - 12kg Weizenmehl
 - 8kg Wasser

Gewinn pro Kiste: 20 Euro



Mehrkornbrot

Zutaten für eine Kiste Mehrkornbrot

- - 6kg Weizenmehl
 - 12kg Wasser
 - 10kg Mischkornschrot

Gewinn pro Kiste: 60 Euro

■ Der Bäcker möchte viel Geld verdienen!

Beispiel: Einfache Bäckerrechnung



Weizenmischbrot

- Zutaten für eine Kiste Weizenmischbrot:
 - 12kg Weizenmehl
 - 8kg Wasser

Gewinn pro Kiste: 20 Euro



Mehrkornbrot

- Zutaten für eine Kiste Mehrkornbrot
 - 6kg Weizenmehl
 - 12kg Wasser
 - 10kg Mischkornschrot

Gewinn pro Kiste: 60 Euro

- **Der Bäcker möchte viel Geld verdienen!**

Beispiel: Einfache Bäckerrechnung

	<i>Weizenmehl</i>	<i>Wasser</i>	<i>Mischkornschrot</i>
Weizenmischbrot	12 kg	8 kg	0 kg
Mehrkornbrot	6 kg	12 kg	10 kg
Kontingent	630 kg	620 kg	350 kg

10 Kisten Weizenmischbrote sind für Stammkunden reserviert!

$$\text{Gewinn} = 20\text{Euro} \cdot \text{Kisten Weizenmischbrot} \\ + 60\text{Euro} \cdot \text{Kisten Mehrkornbrot}$$

Beispiel: Einfache Bäckerrechnung

	<i>Weizenmehl</i>	<i>Wasser</i>	<i>Mischkornschrot</i>
Weizenmischbrot	12 kg	8 kg	0 kg
Mehrkornbrot	6 kg	12 kg	10 kg
Kontingent	630 kg	620 kg	350 kg

10 Kisten Weizenmischbrote sind für Stammkunden reserviert!

Gewinn = 20Euro · Kisten Weizenmischbrot
+ 60Euro · Kisten Mehrkornbrot

Beispiel: Einfache Bäckerrechnung

	<i>Weizenmehl</i>	<i>Wasser</i>	<i>Mischkornschrot</i>
Weizenmischbrot	12 kg	8 kg	0 kg
Mehrkornbrot	6 kg	12 kg	10 kg
Kontingent	630 kg	620 kg	350 kg

10 Kisten Weizenmischbrote sind für Stammkunden reserviert!

$$\begin{aligned} \text{Gewinn} &= 20\text{Euro} \cdot \text{Kisten Weizenmischbrot} \\ &+ 60\text{Euro} \cdot \text{Kisten Mehrkornbrot} \end{aligned}$$

Beispiel: Einfache Bäckerrechnung

	<i>Weizenmehl</i>	<i>Wasser</i>	<i>Mischkornschrot</i>
Weizenmischbrot	12 kg	8 kg	0 kg
Mehrkornbrot	6 kg	12 kg	10 kg
Kontingent	630 kg	620 kg	350 kg

10 Kisten Weizenmischbrote sind für Stammkunden reserviert!

$$\begin{aligned} \text{Gewinn} &= 20\text{Euro} \cdot \text{Kisten Weizenmischbrot} \\ &+ 60\text{Euro} \cdot \text{Kisten Mehrkornbrot} \end{aligned}$$

Beispiel: Einfache Bäckersrechnung

Seien x_1 = Kisten Weizenmischbrot, x_2 = Kisten Mehrkornbrot:

Zielfunktion **ZF:** $f(x_1, x_2) = 20x_1 + 60x_2 = \max!$

Nebenbedingungen **NB:** $12x_1 + 6x_2 \leq 630$

$$8x_1 + 12x_2 \leq 620$$

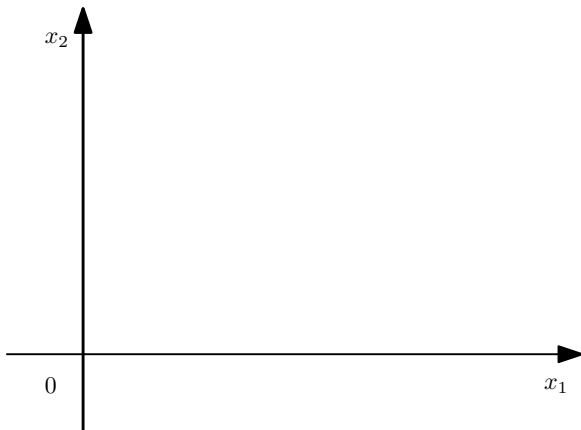
$$10x_2 \leq 350$$

$$x_1 \geq 10$$

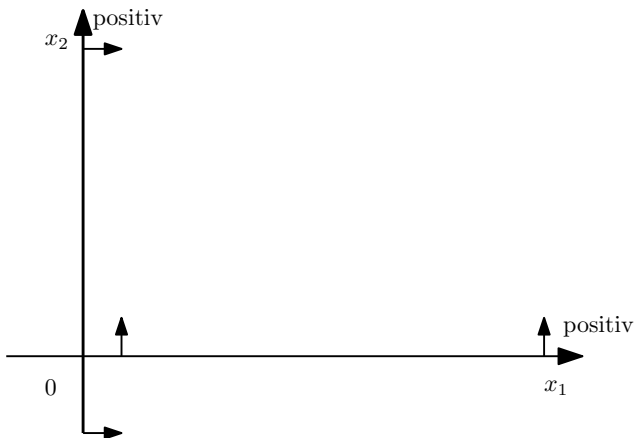
$$x_1 \geq 0$$

$$x_2 \geq 0$$

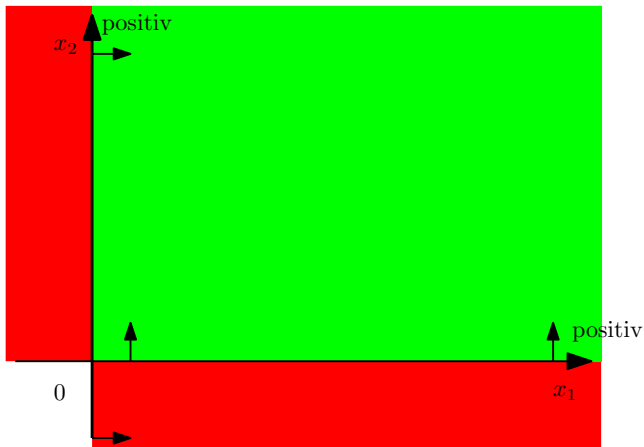
Beispiel: Einfache Bäckerrechnung



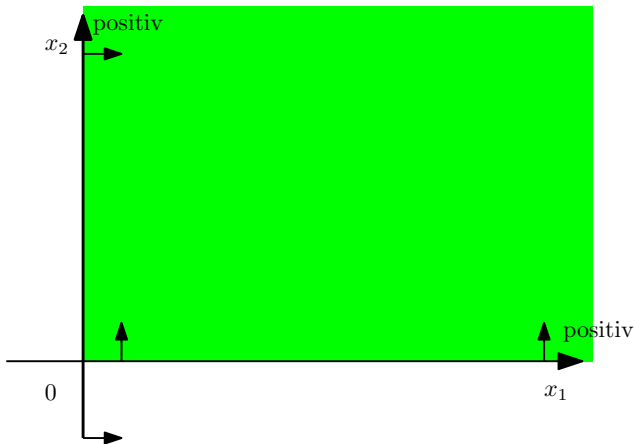
Beispiel: Einfache Bäckerrechnung



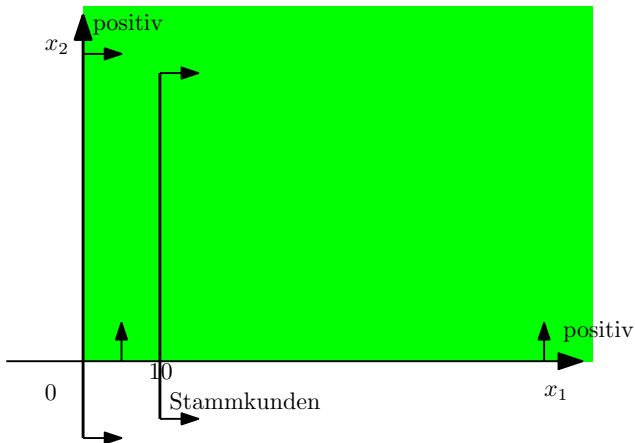
Beispiel: Einfache Bäckerrechnung



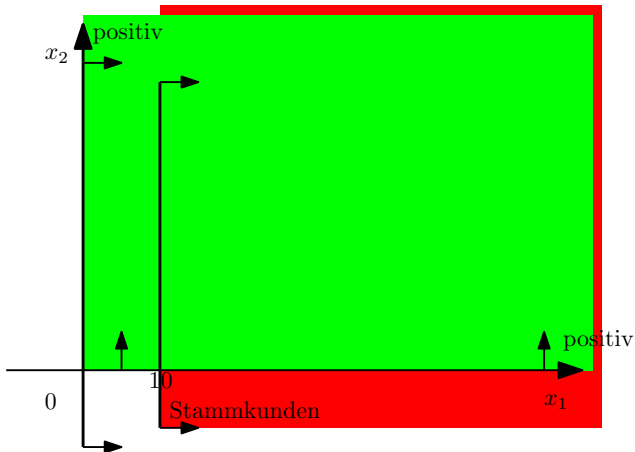
Beispiel: Einfache Bäckersrechnung



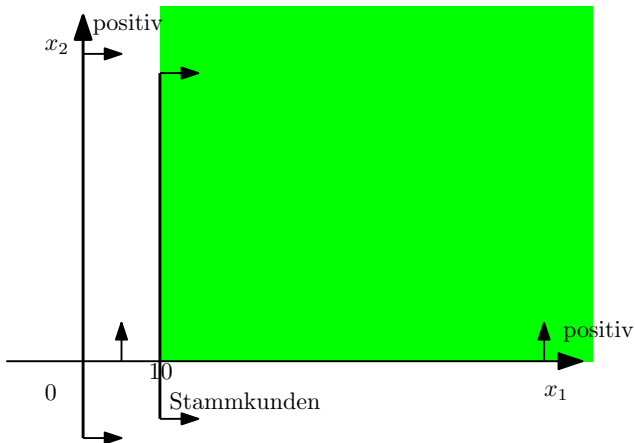
Beispiel: Einfache Bäckerrechnung



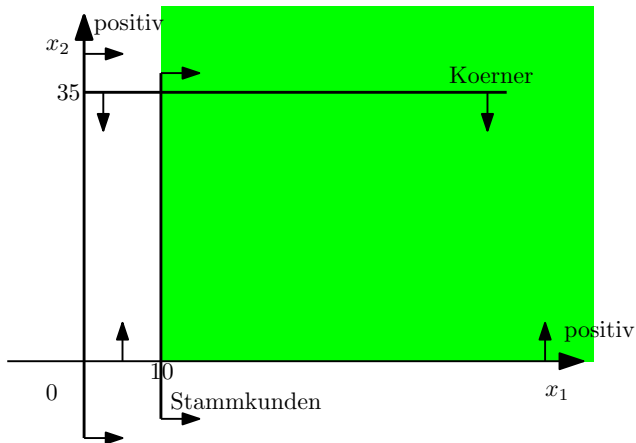
Beispiel: Einfache Bäckerrechnung



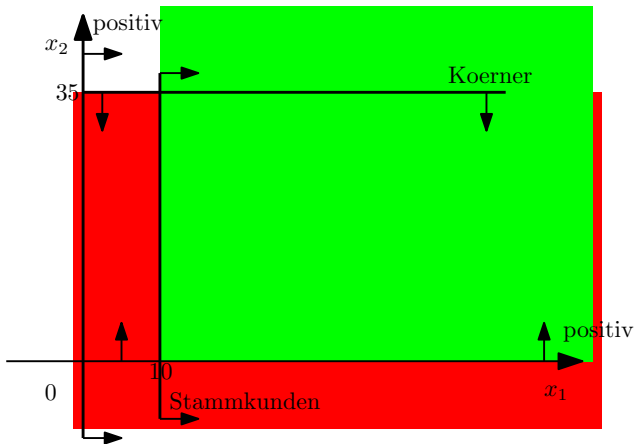
Beispiel: Einfache Bäckerrechnung



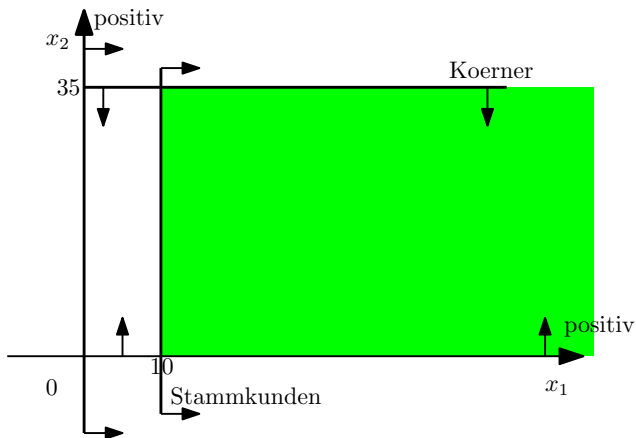
Beispiel: Einfache Bäckerrechnung



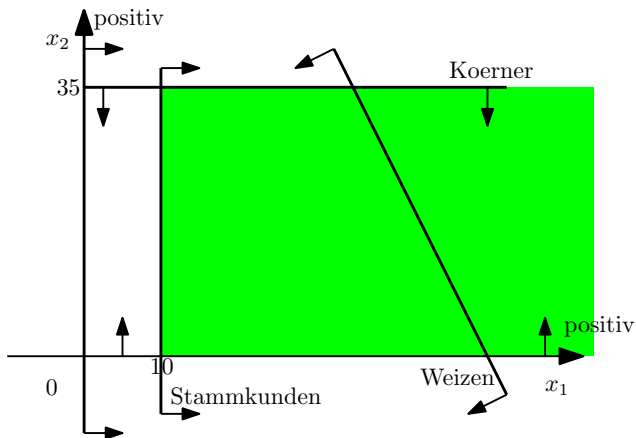
Beispiel: Einfache Bäckerrechnung



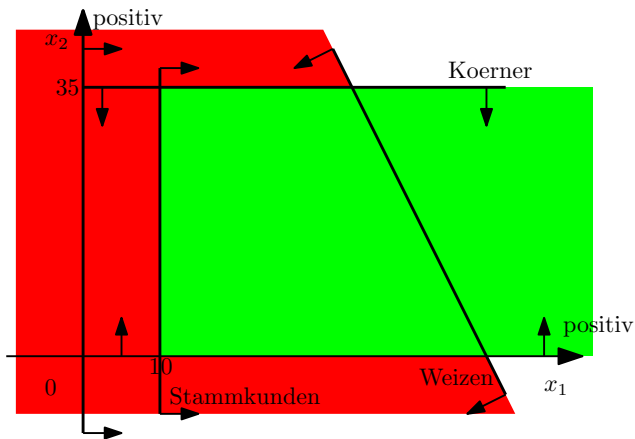
Beispiel: Einfache Bäckerrechnung



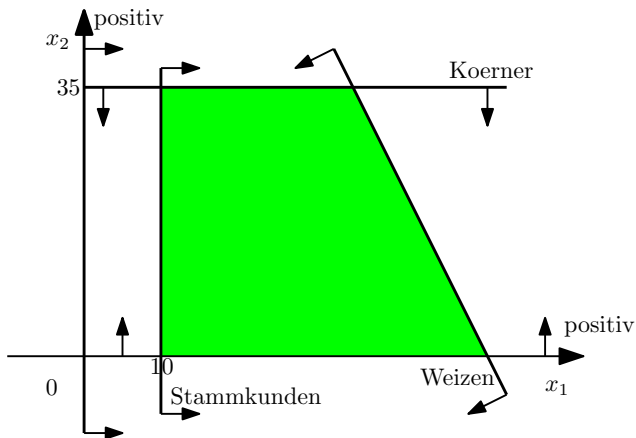
Beispiel: Einfache Bäckerrechnung



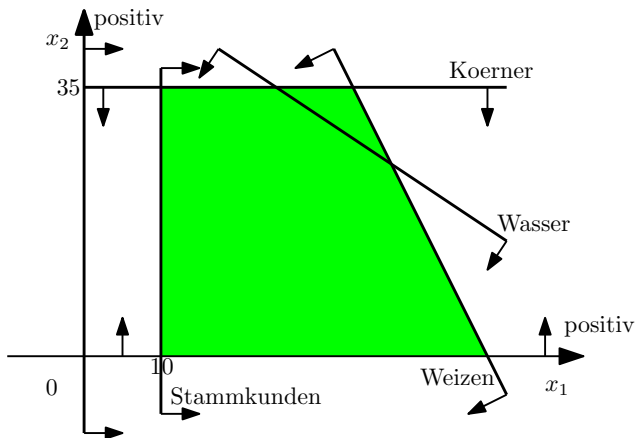
Beispiel: Einfache Bäckerrechnung



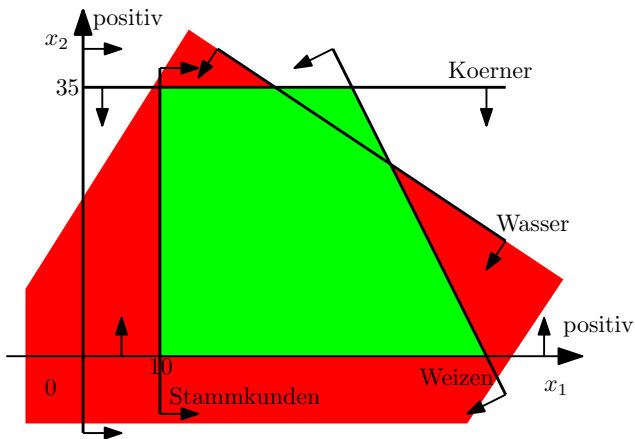
Beispiel: Einfache Bäckerrechnung



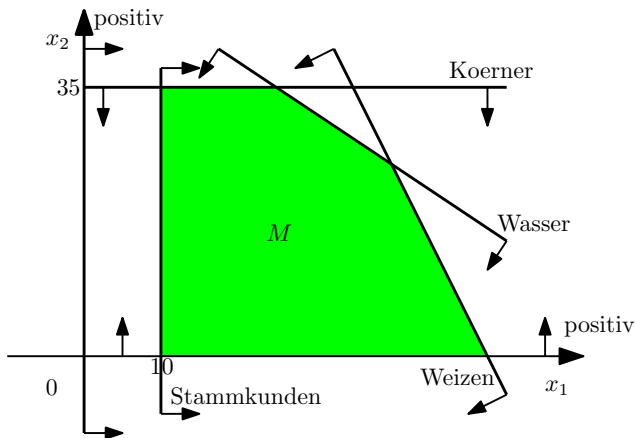
Beispiel: Einfache Bäckerrechnung



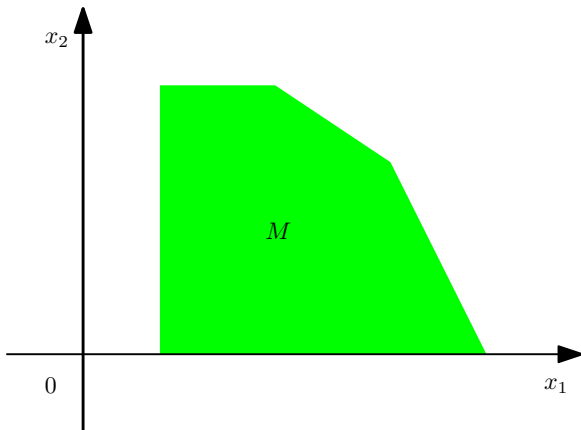
Beispiel: Einfache Bäckerrechnung



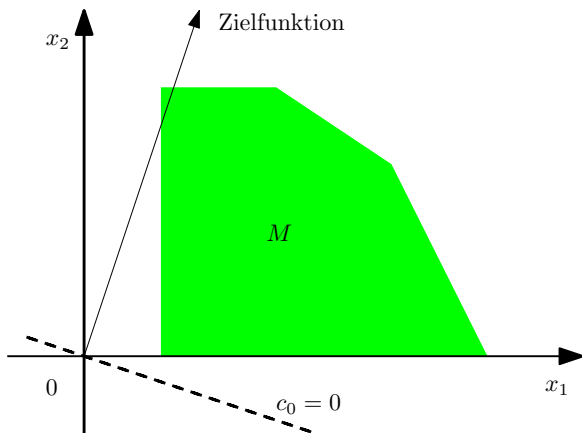
Beispiel: Einfache Bäckerrechnung



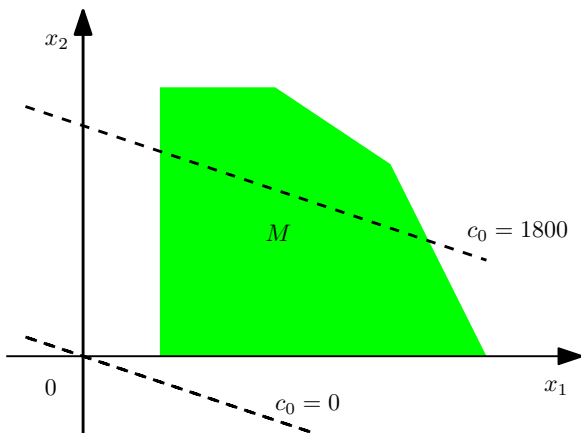
Beispiel: Einfache Bäckersrechnung



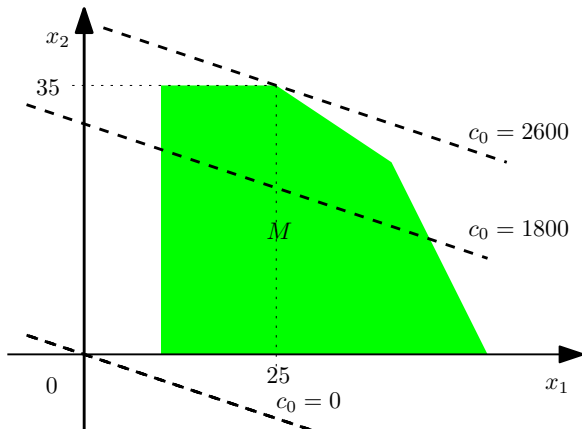
Beispiel: Einfache Bäckersrechnung



Beispiel: Einfache Bäckerrechnung



Beispiel: Einfache Bäckerrechnung



- *Freie Variablen* \implies Differenz zweier nichtnegativer Variablen:

$$x_k = x_k^1 - x_k^2, \quad \text{mit } x_k^1 \geq 0, x_k^2 \geq 0$$

- Ungleichungsbedingungen \implies Addition (oder Subtraktion) nichtnegativer *Schlupfvariablen* \implies Gleichungsbedingungen:

$$a_1 x_1 + \cdots + a_n x_n \leq b$$

\Leftrightarrow

$$a_1 x_1 + \cdots + a_n x_n + x_{n+1} = b, x_{n+1} \geq 0$$

- Freie Variablen \implies Differenz zweier nichtnegativer Variablen:

$$x_k = x_k^1 - x_k^2, \quad \text{mit } x_k^1 \geq 0, x_k^2 \geq 0$$

- Ungleichungsbedingungen \implies Addition (oder Subtraktion) nichtnegativer *Schlupfvariablen* \implies Gleichungsbedingungen:

$$a_1 x_1 + \cdots + a_n x_n \leq b$$

\Leftrightarrow

$$a_1 x_1 + \cdots + a_n x_n + x_{n+1} = b, x_{n+1} \geq 0$$

Beispiel: Einfache Bäckersrechnung

Normalform der linearen Optimierungsaufgabe

$$\mathbf{ZF:} \quad f(\vec{x}) = 20x_1 + 60x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 = \max!$$

$$\mathbf{NB:} \quad 12x_1 + 6x_2 + x_3 = 630$$

$$8x_1 + 12x_2 + x_4 = 620$$

$$10x_2 + x_5 = 350$$

$$x_1 - x_6 = 10$$

$$x_i \geq 0$$

$$\mathbf{ZF:} \quad f(\vec{x}) = c_1 x_1 \dots c_{n-m} x_{n-m} + c_0 \quad = \max!$$

$$\mathbf{NB:} \quad a_{1,1} x_1 + \dots + a_{1,n-m} x_{n-m} + x_{n-m+1} \quad = b_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \dots \quad \quad \quad \vdots$$

$$a_{m,1} x_m + \dots + a_{m,n-m} x_{n-m} \quad + x_n = b_m$$

$$x_i \geq 0$$

- **Zulässigkeitsbereich M** : alle Punkte, die **NB** erfüllen
- M ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist M zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von M
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in M
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke k können m Spaltenvektoren von A zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von k
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von A sind l.u.

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist *M* zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von *M*
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in *M*
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke *k* können *m* Spaltenvektoren von *A* zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von *k*
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von *A* sind l.u.

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist *M* zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von *M*
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in *M*
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke *k* können *m* Spaltenvektoren von *A* zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von *k*
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von *A* sind l.u.

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist *M* zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von *M*
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in *M*
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke *k* können *m* Spaltenvektoren von *A* zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von *k*
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von *A* sind l.u.

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist *M* zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von *M*
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in *M*
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke *k* können *m* Spaltenvektoren von *A* zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von *k*
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von *A* sind l.u.

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist *M* zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von *M*
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in *M*
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke *k* können *m* Spaltenvektoren von *A* zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von *k*
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von *A* sind l.u.

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist *M* zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von *M*
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in *M*
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke *k* können *m* Spaltenvektoren von *A* zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von *k*
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von *A* sind l.u.

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist *M* zusätzlich beschränkt: *Polytop*
- *Seite*: nichtleere, beschränkende Halbebene, + Schnitte
- *Extremalstrahl*: Halbgerade und 1-Seite von *M*
- *Ecke*: $\vec{x} \in M$ und keine Konvexkombination in *M*
($\forall \vec{x}_1 \neq \vec{x}_2 \in M : \vec{x} \neq \lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2, 0 < \lambda < 1$)
- *Basis einer Ecke*: Jeder Ecke *k* können *m* Spaltenvektoren von *A* zugeordnet werden
diese korrespondieren zu den Einträgen $\neq 0$ von *k*
- \vec{x} ist Ecke \Leftrightarrow die entspr. Spalten von *A* sind l.u.

Lösbarkeit von Linearen Programmen

(LP)

$$\begin{aligned} f(x) = \langle x, p \rangle &= \max! \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

$$M = \{Ax \leq b, x \geq 0\}$$

Unlösbar falls:

- $M = \emptyset$
- $\sup_{x \in M} f(x) = \infty$

Sei $M \neq \emptyset$ und f auf M nach oben beschränkt \Rightarrow

- LP ist lösbar
- Menge der Lösungen ist eine ℓ -Seite
- mind. eine Ecke ist Lösung

Lösbarkeit von Linearen Programmen

(LP)

$$\begin{aligned} f(x) = \langle x, p \rangle &= \max! \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

$$M = \{Ax \leq b, x \geq 0\}$$

Unlösbar falls:

- $M = \emptyset$
- $\sup_{x \in M} f(x) = \infty$

Sei $M \neq \emptyset$ und f auf M nach oben beschränkt \Rightarrow

- LP ist lösbar
- Menge der Lösungen ist eine ℓ -Seite
- mind. eine Ecke ist Lösung

Lösbarkeit von Linearen Programmen

(LP)

$$\begin{aligned} f(x) = \langle x, p \rangle &= \max! \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

$$M = \{Ax \leq b, x \geq 0\}$$

Unlösbar falls:

- $M = \emptyset$
- $\sup_{x \in M} f(x) = \infty$

Sei $M \neq \emptyset$ und f auf M nach oben beschränkt \Rightarrow

- LP ist lösbar
- Menge der Lösungen ist eine ℓ -Seite
- mind. eine Ecke ist Lösung

Lösbarkeit von Linearen Programmen

(LP)

$$\begin{aligned} f(x) = \langle x, p \rangle &= \max! \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

$$M = \{Ax \leq b, x \geq 0\}$$

Unlösbar falls:

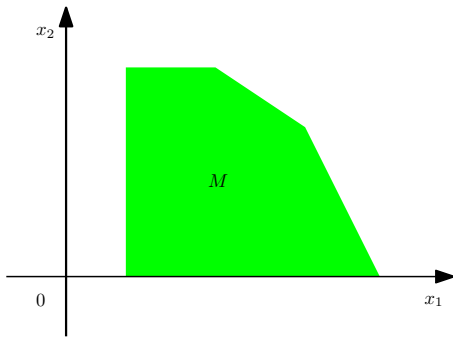
- $M = \emptyset$
- $\sup_{x \in M} f(x) = \infty$

Sei $M \neq \emptyset$ und f auf M nach oben beschränkt \Rightarrow

- LP ist lösbar
- Menge der Lösungen ist eine ℓ -Seite
- mind. eine Ecke ist Lösung

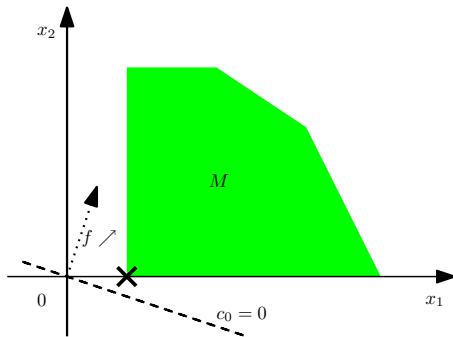
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung.
- Beliebige Startecke x .
- Suche Nachbarecke \bar{x} mit $f(\bar{x}) > f(x)$.
- Bis man keine mehr findet \Rightarrow Optimum



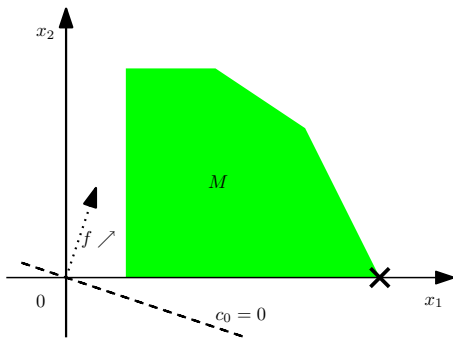
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung. Beliebige Startecke x . Suche Nachbarecke \bar{x} mit $f(\bar{x}) > f(x)$. Bis man keine mehr findet \Rightarrow Optimum



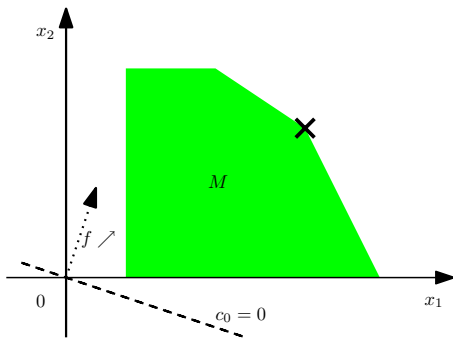
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung. Beliebige Startecke x . Suche Nachbarecke \bar{x} mit $f(\bar{x}) > f(x)$. Bis man keine mehr findet \Rightarrow Optimum



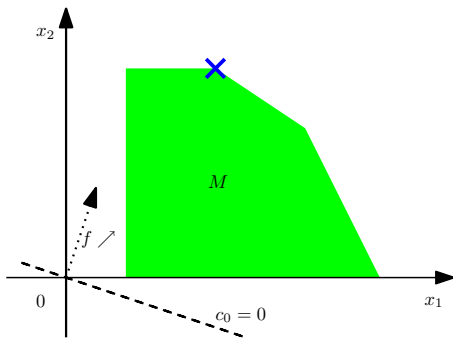
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung. Beliebige Startecke x . Suche Nachbarecke \bar{x} mit $f(\bar{x}) > f(x)$. Bis man keine mehr findet \Rightarrow Optimum



Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung.
- Beliebige Startecke x .
- Suche Nachbarecke \bar{x} mit $f(\bar{x}) > f(x)$.
- Bis man keine mehr findet \Rightarrow Optimum



- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
 - ① Keine weitere Verbesserung ist möglich \Rightarrow Optimum
 - ② Es existiert eine Richtung, in der f unbeschränkt wächst \Rightarrow keine Lösung
 - ③ Verbesserung möglich \Rightarrow weiter
- Gauss-artige Umwandlung des Tableaus um gefundenes Pivotelement
- Nochmal!

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
 - ① Keine weitere Verbesserung ist möglich \Rightarrow Optimum
 - ② Es existiert eine Richtung, in der f unbeschränkt wächst \Rightarrow keine Lösung
 - ③ Verbesserung möglich \Rightarrow weiter
- Gauss-artige Umwandlung des Tableaus um gefundenes Pivotelement
- Nochmal!

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
 - ① Keine weitere Verbesserung ist möglich \Rightarrow Optimum
 - ② Es existiert eine Richtung, in der f unbeschränkt wächst \Rightarrow keine Lösung
 - ③ Verbesserung möglich \Rightarrow weiter
- Gauss-artige Umwandlung des Tableaus um gefundenes Pivotelement
- Nochmal!

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
 - ① Keine weitere Verbesserung ist möglich \Rightarrow Optimum
 - ② Es existiert eine Richtung, in der f unbeschränkt wächst \Rightarrow keine Lösung
 - ③ Verbesserung möglich \Rightarrow weiter
- Gauss-artige Umwandlung des Tableaus um gefundenes Pivotelement
- Nochmal!

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
 - ① Keine weitere Verbesserung ist möglich \Rightarrow Optimum
 - ② Es existiert eine Richtung, in der f unbeschränkt wächst \Rightarrow keine Lösung
 - ③ Verbesserung möglich \Rightarrow weiter
- Gauss-artige Umwandlung des Tableaus um gefundenes Pivotelement
- Nochmal!

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
 - ① Keine weitere Verbesserung ist möglich \Rightarrow Optimum
 - ② Es existiert eine Richtung, in der f unbeschränkt wächst \Rightarrow keine Lösung
 - ③ Verbesserung möglich \Rightarrow weiter
- Gauss-artige Umwandlung des Tableaus um gefundenes Pivotelement
- Nochmal!

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
 - ① Keine weitere Verbesserung ist möglich \Rightarrow Optimum
 - ② Es existiert eine Richtung, in der f unbeschränkt wächst \Rightarrow keine Lösung
 - ③ Verbesserung möglich \Rightarrow weiter
- Gauss-artige Umwandlung des Tableaus um gefundenes Pivotelement
- Nochmal!

Probleme bei praktischen Berechnung

- **Keine Startecke bekannt.** Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, \Rightarrow Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...
- **Entartete Ecken.** Störung des Systems durch $\vec{b}^1 = \vec{b} + \vec{e}_i$
- **Langsam.** Verschiedene Suchmethoden beim Eckenwechsel (Pivotsuche)

Probleme bei praktischen Berechnung

- **Keine Startecke bekannt.** Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, \Rightarrow Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...
- **Entartete Ecken.** Störung des Systems durch $\vec{b}^1 = \vec{b} + \vec{\epsilon}$
- **Langsam.** Verschiedene Suchmethoden beim Eckenwechsel (Pivotsuche)

Probleme bei praktischen Berechnung

- **Keine Startecke bekannt.** Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, \Rightarrow Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...
- **Entartete Ecken.** Störung des Systems durch $\vec{b}^1 = \vec{b} + \vec{\varepsilon}$
- **Langsam.** Verschiedene Suchmethoden beim Eckenwechsel (Pivotsuche)

Probleme bei praktischen Berechnung

- **Keine Startecke bekannt.** Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, \Rightarrow Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...
- **Entartete Ecken.** Störung des Systems durch $\vec{b}^1 = \vec{b} + \vec{\varepsilon}$
- **Langsam.** Verschiedene Suchmethoden beim Eckenwechsel (Pivotsuche)

Probleme bei praktischen Berechnung

- **Keine Startecke bekannt.** Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, \Rightarrow Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...
- **Entartete Ecken.** Störung des Systems durch $\vec{b}^1 = \vec{b} + \vec{\varepsilon}$
- **Langsam.** Verschiedene Suchmethoden beim Eckenwechsel (Pivotsuche)

Probleme bei praktischen Berechnung

- **Keine Startecke bekannt.** Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, \Rightarrow Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...
- **Entartete Ecken.** Störung des Systems durch $\vec{b}^1 = \vec{b} + \vec{\varepsilon}$
- **Langsam.** Verschiedene Suchmethoden beim Eckenwechsel (Pivotsuche)

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPSolve* (Java, frei)

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPSolve* (Java, frei)

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPSolve* (Java, frei)

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPSolve* (Java, frei)

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPSolve* (Java, frei)

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPsolve* (Java, frei)

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPsolve* (Java, frei)

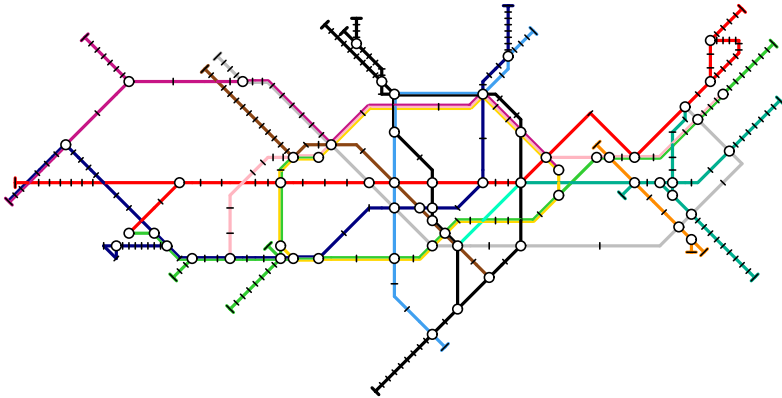
- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- Layout-Probleme (Graphentheorie)
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs!
(ca. 1/3)
- Software: *CPLEX*, *XPRESS*, *LPSolve* (Java, frei)

Automatisches Zeichnen von U-Bahn Plänen



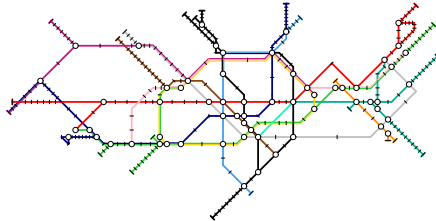
Das Londoner U-Bahn Netz geographisch

Automatisches Zeichnen von U-Bahn Plänen



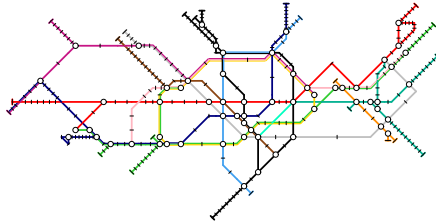
Londoner U-Bahn schematisch, oktilinear, lesbar, topologietreu
Zielfunktion (u.a.): 'Geradlinigkeit', 'Relative Positionen', und 'Kurze Gesamtkantenlänge'.

Automatisches Zeichnen von U-Bahn Plänen



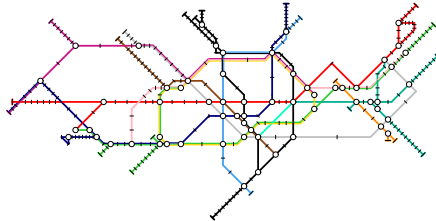
- $|V| = 298$ Knoten, $|E| = 340$ Kanten
- effektive Laufzeit (2.2 GHz Opteron 16GB Ram) 15 min.
- 20447 Variablen
- Ursprünglich ca. 1 Mio. Constraints
- Davon fast 1 Mio. Planarität; 11000 andere
- manuell/heuristische Reduktion auf 11000+66000
- Mit sukzessivem *Warmstart* 11089 Constraints

Automatisches Zeichnen von U-Bahn Plänen



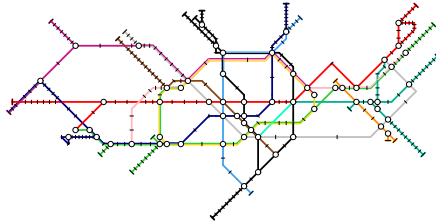
- $|V| = 298$ Knoten, $|E| = 340$ Kanten
- effektive Laufzeit (2.2 GHz Opteron 16GB Ram) 15 min.
- 20447 Variablen
- Ursprünglich ca. 1 Mio. Constraints
- Davon fast 1 Mio. Planarität; 11000 andere
- manuell/heuristische Reduktion auf 11000+66000
- Mit sukzessivem *Warmstart* 11089 Constraints

Automatisches Zeichnen von U-Bahn Plänen



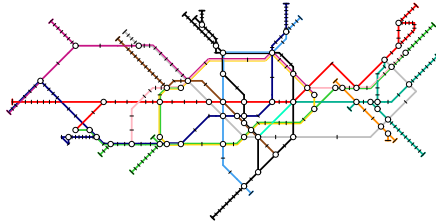
- $|V| = 298$ Knoten, $|E| = 340$ Kanten
- effektive Laufzeit (2.2 GHz Opteron 16GB Ram) 15 min.
- 20447 Variablen
- Ursprünglich ca. 1 Mio. Constraints
- Davon fast 1 Mio. Planarität; 11000 andere
- manuell/heuristische Reduktion auf 11000+66000
- Mit sukzessivem *Warmstart* 11089 Constraints

Automatisches Zeichnen von U-Bahn Plänen



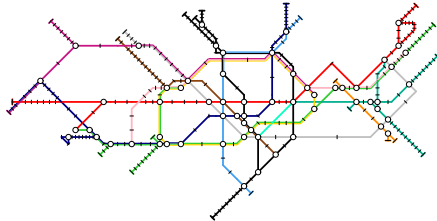
- $|V| = 298$ Knoten, $|E| = 340$ Kanten
- effektive Laufzeit (2.2 GHz Opteron 16GB Ram) 15 min.
- 20447 Variablen
- Ursprünglich ca. 1 Mio. Constraints
- Davon fast 1 Mio. Planarität; 11000 andere
- manuell/heuristische Reduktion auf 11000+66000
- Mit sukzessivem *Warmstart* 11089 Constraints

Automatisches Zeichnen von U-Bahn Plänen



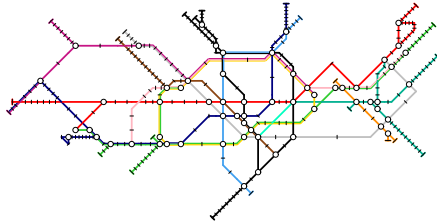
- $|V| = 298$ Knoten, $|E| = 340$ Kanten
- effektive Laufzeit (2.2 GHz Opteron 16GB Ram) 15 min.
- 20447 Variablen
- Ursprünglich ca. 1 Mio. Constraints
- Davon fast 1 Mio. Planarität; 11000 andere
- manuell/heuristische Reduktion auf 11000+66000
- Mit sukzessivem *Warmstart* 11089 Constraints

Automatisches Zeichnen von U-Bahn Plänen



- $|V| = 298$ Knoten, $|E| = 340$ Kanten
- effektive Laufzeit (2.2 GHz Opteron 16GB Ram) 15 min.
- 20447 Variablen
- Ursprünglich ca. 1 Mio. Constraints
- Davon fast 1 Mio. Planarität; 11000 andere
- manuell/heuristische Reduktion auf 11000+66000
- Mit sukzessivem *Warmstart* 11089 Constraints

Automatisches Zeichnen von U-Bahn Plänen

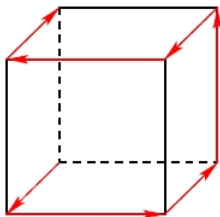


- $|V| = 298$ Knoten, $|E| = 340$ Kanten
- effektive Laufzeit (2.2 GHz Opteron 16GB Ram) 15 min.
- 20447 Variablen
- Ursprünglich ca. 1 Mio. Constraints
- Davon fast 1 Mio. Planarität; 11000 andere
- manuell/heuristische Reduktion auf 11000+66000
- Mit sukzessivem *Warmstart* 11089 Constraints

- Zunächst: exponentiell
- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

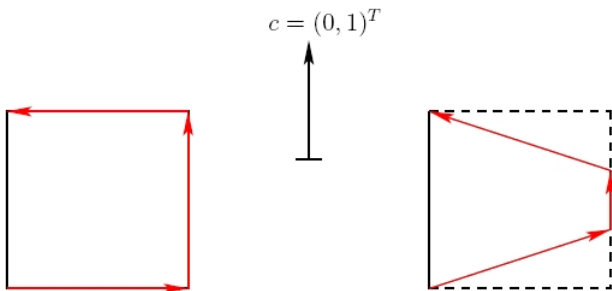
- Zunächst: exponentiell
- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

- Zunächst: exponentiell



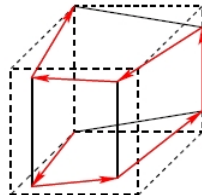
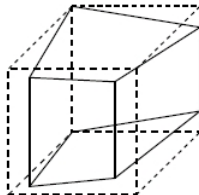
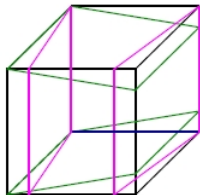
- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

- Zunächst: exponentiell



- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

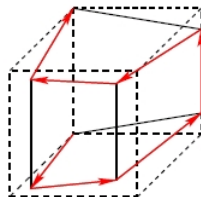
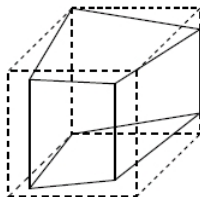
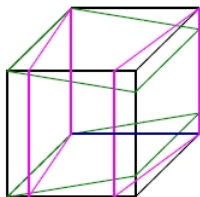
- Zunächst: exponentiell



(Klee-Minty-Polyeder)

- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

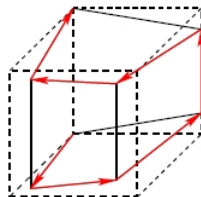
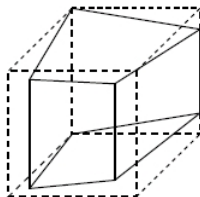
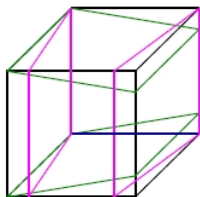
- Zunächst: exponentiell



(Klee-Minty-Polyeder)

- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

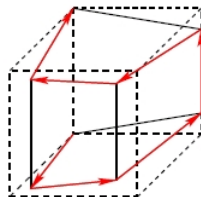
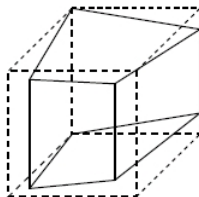
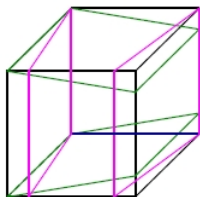
- Zunächst: exponentiell



(Klee-Minty-Polyeder)

- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

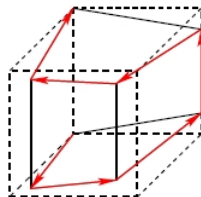
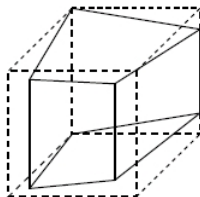
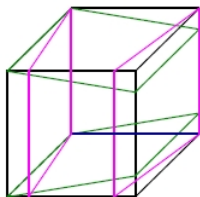
- Zunächst: exponentiell



(Klee-Minty-Polyeder)

- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

- Zunächst: exponentiell



(Klee-Minty-Polyeder)

- \exists ähnliche Beispiele für die meisten Varianten
- \exists Varianten: exponentielle Instanz unbekannt
- \Rightarrow Nicht bekannt ob Simplex polynomial ist
- Durchschnittlich: $2m-3m$ Schritte ($m = \#$ Gleichungen)

Theorem (Khachian 1979)

LP können in polynomialer Zeit gelöst werden.

- K. zeigte, dass der von Shor 1977 vorgestellte *Ellipsoid-Algorithmus* polynomiale Laufzeit hat.
- Realität: dramatisch langsamer.
- N. Karmakar (1984): Anderer (nicht SA-basierter) polynomialer Algorithmus für LP: *innere-Punkt-Methode*.
- Innerer-Punkt-Algorithmus praktisch konkurrenzfähig mit Simplex

Theorem (Khachian 1979)

LP können in polynomialer Zeit gelöst werden.

- K. zeigte, dass der von Shor 1977 vorgestellte *Ellipsoid-Algorithmus* polynomiale Laufzeit hat.
- Realität: dramatisch langsamer.
- N. Karmakar (1984): Anderer (nicht SA-basierter) polynomialer Algorithmus für LP: *innere-Punkt-Methode*.
- Innerer-Punkt-Algorithmus praktisch konkurrenzfähig mit Simplex

Theorem (Khachian 1979)

LP können in polynomialer Zeit gelöst werden.

- K. zeigte, dass der von Shor 1977 vorgestellte *Ellipsoid-Algorithmus* polynomiale Laufzeit hat.
- Realität: dramatisch langsamer.
- N. Karmakar (1984): Anderer (nicht SA-basierter) polynomialer Algorithmus für LP: *innere-Punkt-Methode*.
- Innerer-Punkt-Algorithmus praktisch konkurrenzfähig mit Simplex

Theorem (Khachian 1979)

LP können in polynomialer Zeit gelöst werden.

- K. zeigte, dass der von Shor 1977 vorgestellte *Ellipsoid-Algorithmus* polynomiale Laufzeit hat.
- Realität: dramatisch langsamer.
- N. Karmakar (1984): Anderer (nicht SA-basierter) polynomialer Algorithmus für LP: *innere-Punkt-Methode*.
- Innerer-Punkt-Algorithmus praktisch konkurrenzfähig mit Simplex

Theorem (Khachian 1979)

LP können in polynomialer Zeit gelöst werden.

- K. zeigte, dass der von Shor 1977 vorgestellte *Ellipsoid-Algorithmus* polynomiale Laufzeit hat.
- Realität: dramatisch langsamer.
- N. Karmakar (1984): Anderer (nicht SA-basierter) polynomialer Algorithmus für LP: *innere-Punkt-Methode*.
- Innerer-Punkt-Algorithmus praktisch konkurrenzfähig mit Simplex

Lemma (Farkas)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Dann gilt genau eine der folgenden Aussagen:

- 1 $Ax = b, x \geq 0$ ist lösbar durch ein $x \in \mathbb{R}^n$
- 2 $A^T y \leq 0, b^T y > 0$ ist lösbar durch ein $y \in \mathbb{R}^m$

Beweis.

- 1 und 2 $\Rightarrow 0 < y^T b = y^T Ax = (A^T y)^T x \leq 0$, W.!
- 2 $\Rightarrow 1 \Rightarrow b \notin K := \{Ax : x \in \mathbb{R}^n, x \geq 0\}$.



Lemma (Farkas)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Dann gilt genau eine der folgenden Aussagen:

- 1 $Ax = b, x \geq 0$ ist lösbar durch ein $x \in \mathbb{R}^n$
- 2 $A^T y \leq 0, b^T y > 0$ ist lösbar durch ein $y \in \mathbb{R}^m$

Beweis.

1 und 2 $\Rightarrow 0 < y^T b = y^T Ax = (A^T y)^T x \leq 0$, W.!

2 $\Rightarrow b \notin K := \{Ax : x \in \mathbb{R}^n, x \geq 0\}$.

K ist ein Polyeder (Weyl) $\Rightarrow K$ abgeschlossen

\Rightarrow (Trennungssatz) $\exists y \in \mathbb{R}^m, y \neq 0$, so dass

$y^T Ax \leq 0 < y^T b, \forall x \geq 0$.

Setze $x = e_i \Rightarrow y^T A \leq 0 \Rightarrow 2$



Lemma (Farkas)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Dann gilt genau eine der folgenden Aussagen:

- 1 $Ax = b, x \geq 0$ ist lösbar durch ein $x \in \mathbb{R}^n$
- 2 $A^T y \leq 0, b^T y > 0$ ist lösbar durch ein $y \in \mathbb{R}^m$

Beweis.

- 1 und 2 $\Rightarrow 0 < y^T b = y^T Ax = (A^T y)^T x \leq 0$, W.!
- 2 $\Rightarrow b \notin K := \{Ax : x \in \mathbb{R}^n, x \geq 0\}$.

K ist ein Polyeder (Weyl) $\Rightarrow K$ abgeschlossen

\Rightarrow (Trennungssatz) $\exists y \in \mathbb{R}^m, y \neq 0$, so dass

$y^T Ax \leq 0 < y^T b, \forall x \geq 0$.

Setze $x = e_i \Rightarrow y^T A \leq 0 \Rightarrow 2$



Lemma (Farkas)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Dann gilt genau eine der folgenden Aussagen:

- 1 $Ax = b, x \geq 0$ ist lösbar durch ein $x \in \mathbb{R}^n$
- 2 $A^T y \leq 0, b^T y > 0$ ist lösbar durch ein $y \in \mathbb{R}^m$

Beweis.

- 1 und 2 $\Rightarrow 0 < y^T b = y^T Ax = (A^T y)^T x \leq 0$, W.!
- 2 $\Rightarrow b \notin K := \{Ax : x \in \mathbb{R}^n, x \geq 0\}$.
 K ist ein Polyeder (Weyl) $\Rightarrow K$ abgeschlossen
 \Rightarrow (Trennungssatz) $\exists y \in \mathbb{R}^m, y \neq 0$, so dass
 $y^T Ax \leq 0 < y^T b, \forall x \geq 0$.
Setze $x = e_i \Rightarrow y^T A \leq 0 \Rightarrow 2$



Lemma (Farkas)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Dann gilt genau eine der folgenden Aussagen:

- 1 $Ax = b, x \geq 0$ ist lösbar durch ein $x \in \mathbb{R}^n$
- 2 $A^T y \leq 0, b^T y > 0$ ist lösbar durch ein $y \in \mathbb{R}^m$

Beweis.

- 1 und 2 $\Rightarrow 0 < y^T b = y^T Ax = (A^T y)^T x \leq 0$, W.!
- 2 $\Rightarrow b \notin K := \{Ax : x \in \mathbb{R}^n, x \geq 0\}$.
 K ist ein Polyeder (Weyl) $\Rightarrow K$ abgeschlossen
 \Rightarrow (Trennungssatz) $\exists y \in \mathbb{R}^m, y \neq 0$, so dass
 $y^T Ax \leq 0 < y^T b, \forall x \geq 0$.

Setze $x = e_i \Rightarrow y^T A \leq 0 \Rightarrow 2$



Lemma (Farkas)

Seien $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Dann gilt genau eine der folgenden Aussagen:

- 1 $Ax = b, x \geq 0$ ist lösbar durch ein $x \in \mathbb{R}^n$
- 2 $A^T y \leq 0, b^T y > 0$ ist lösbar durch ein $y \in \mathbb{R}^m$

Beweis.

- 1 und 2 $\Rightarrow 0 < y^T b = y^T Ax = (A^T y)^T x \leq 0$, W.!
- 2 $\Rightarrow b \notin K := \{Ax : x \in \mathbb{R}^n, x \geq 0\}$.
 K ist ein Polyeder (Weyl) $\Rightarrow K$ abgeschlossen
 \Rightarrow (Trennungssatz) $\exists y \in \mathbb{R}^m, y \neq 0$, so dass
 $y^T Ax \leq 0 < y^T b, \forall x \geq 0$.
Setze $x = e_i \Rightarrow y^T A \leq 0 \Rightarrow 2$



(PP)

$$f(x) = x^T c = \max!$$

$$Ax \leq b$$

$$x \geq 0$$

$$M = \{Ax \leq b, x \geq 0\}$$

(DP)

$$g(y) = y^T b = \min!$$

$$A^T y \geq c$$

$$y \geq 0$$

$$N = \{A^T y \geq c, y \geq 0\}$$

(PP)

$$f(\vec{x}) = \vec{x}_1^T \vec{p}_1 + \vec{x}_2^T \vec{p}_2 = \max!$$

$$A_{1,1}\vec{x}_1 + A_{1,2}\vec{x}_2 \leq b_1$$

$$A_{2,1}\vec{x}_1 + A_{2,2}\vec{x}_2 = b_2$$

$$\vec{x}_1 \geq 0, \quad \vec{x}_2 \text{ frei}$$

(DP)

$$f(\vec{u}) = \vec{u}_1^T \vec{b}_1 + \vec{u}_2^T \vec{b}_2 = \min!$$

$$A_{1,1}^T \vec{u}_1 + A_{1,2}^T \vec{u}_2 \leq p_1$$

$$A_{2,1}^T \vec{u}_1 + A_{2,2}^T \vec{u}_2 = p_2$$

$$\vec{u}_1 \geq 0, \quad \vec{u}_2 \text{ frei}$$

Theorem

Seien **(PP)** und **(DP)** gegeben.

① Schwacher Dualitätssatz:

- ① $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
- ② $\exists x \in M, u \in N \Rightarrow$ **(PP)** und **(DP)** lösbar.
- ③ $x_0 \in M, u_0 \in N, f(x_0) = g(u_0) \Rightarrow x_0, u_0$ sind Lösungen.

② Starker Dualitätssatz:

- ① **(PP)** lösbar \Rightarrow **(DP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
- ② **(DP)** lösbar \Rightarrow **(PP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Theorem

Seien **(PP)** und **(DP)** gegeben.

① *Schwacher Dualitätssatz:*

- ① $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
- ② $\exists x \in M, u \in N \Rightarrow$ **(PP)** und **(DP)** lösbar.
- ③ $x_0 \in M, u_0 \in N, f(x_0) = g(u_0) \Rightarrow x_0, u_0$ sind Lösungen.

② *Starker Dualitätssatz:*

- ① **(PP)** lösbar \Rightarrow **(DP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
- ② **(DP)** lösbar \Rightarrow **(PP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Theorem

Seien **(PP)** und **(DP)** gegeben.

① *Schwacher Dualitätssatz:*

- ① $x \in M, u \in N \Rightarrow f(x) \leq g(u).$
- ② $\exists x \in M, u \in N \Rightarrow$ **(PP)** und **(DP)** lösbar.
- ③ $x_0 \in M, u_0 \in N, f(x_0) = g(u_0) \Rightarrow x_0, u_0$ sind Lösungen.

② *Starker Dualitätssatz:*

- ① (PP) lösbar \Rightarrow (DP) lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
- ② (DP) lösbar \Rightarrow (PP) lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Theorem

Seien **(PP)** und **(DP)** gegeben.

① *Schwacher Dualitätssatz:*

- ① $x \in M, u \in N \Rightarrow f(x) \leq g(u).$
- ② $\exists x \in M, u \in N \Rightarrow$ **(PP)** und **(DP)** lösbar.
- ③ $x_0 \in M, u_0 \in N, f(x_0) = g(u_0) \Rightarrow x_0, u_0$ sind Lösungen.

② *Starker Dualitätssatz:*

- ① (PP) lösbar \Rightarrow (DP) lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
- ② (DP) lösbar \Rightarrow (PP) lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Theorem

Seien **(PP)** und **(DP)** gegeben.

① *Schwacher Dualitätssatz:*

- ① $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
- ② $\exists x \in M, u \in N \Rightarrow$ **(PP)** und **(DP)** lösbar.
- ③ $x_0 \in M, u_0 \in N, f(x_0) = g(u_0) \Rightarrow x_0, u_0$ sind Lösungen.

② *Starker Dualitätssatz:*

- ① **(PP)** lösbar \Rightarrow **(DP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
- ② **(DP)** lösbar \Rightarrow **(PP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Theorem

Seien **(PP)** und **(DP)** gegeben.

① *Schwacher Dualitätssatz:*

- ① $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
- ② $\exists x \in M, u \in N \Rightarrow$ **(PP)** und **(DP)** lösbar.
- ③ $x_0 \in M, u_0 \in N, f(x_0) = g(u_0) \Rightarrow x_0, u_0$ sind Lösungen.

② *Starker Dualitätssatz:*

- ① **(PP)** lösbar \Rightarrow **(DP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
- ② **(DP)** lösbar \Rightarrow **(PP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Theorem

Seien **(PP)** und **(DP)** gegeben.

① *Schwacher Dualitätssatz:*

- ① $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
- ② $\exists x \in M, u \in N \Rightarrow$ **(PP)** und **(DP)** lösbar.
- ③ $x_0 \in M, u_0 \in N, f(x_0) = g(u_0) \Rightarrow x_0, u_0$ sind Lösungen.

② *Starker Dualitätssatz:*

- ① **(PP)** lösbar \Rightarrow **(DP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
- ② **(DP)** lösbar \Rightarrow **(PP)** lösbar
und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

- Normalform für ein duales Problem leichter zu finden
- # Restriktionen \gg # Variablen \Rightarrow Verringerung des Rechenaufwandes im Dualen

- Normalform für ein duales Problem leichter zu finden
- # Restriktionen \gg # Variablen \Rightarrow Verringerung des Rechenaufwandes im Dualen

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)
- *Relaxierung* $ILP \rightarrow LP$ (ohne Ganzzahligkeitsbedingung)
Dann: Durch Runden (suboptimale) ganzzahlige Lösung finden
- Diese kann u.U. nicht existieren ($\notin M$)
- Wann kann man $ILP \rightarrow LP$ reduzieren?

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)
- *Relaxierung* $ILP \rightarrow LP$ (ohne Ganzzahligkeitsbedingung)
Dann: Durch Runden (suboptimale) ganzzahlige Lösung finden
- Diese kann u.U. nicht existieren ($\notin M$)
- Wann kann man $ILP \rightarrow LP$ reduzieren?

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)
- *Relaxierung* $ILP \rightarrow LP$ (ohne Ganzzahligkeitsbedingung)
Dann: Durch Runden (suboptimale) ganzzahlige Lösung finden
- Diese kann u.U. nicht existieren ($\notin M$)
- Wann kann man $ILP \rightarrow LP$ reduzieren?

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)
- *Relaxierung* $ILP \rightarrow LP$ (ohne Ganzzahligkeitsbedingung)
Dann: Durch Runden (suboptimale) ganzzahlige Lösung finden
- Diese kann u.U. nicht existieren ($\notin M$)
- Wann kann man $ILP \rightarrow LP$ reduzieren?

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)
- *Relaxierung* $ILP \rightarrow LP$ (ohne Ganzzahligkeitsbedingung)
Dann: Durch Runden (suboptimale) ganzzahlige Lösung finden
- Diese kann u.U. nicht existieren ($\notin M$)
- Wann kann man $ILP \rightarrow LP$ reduzieren?

Definition

Matrix A total unimodular falls jede quadratische Submatrix Determinante aus $\{-1, 0, 1\}$ hat.

NB: Die Elemente von A sind alle aus $\{-1, 0, 1\}$

Theorem

LP: $Ax \leq b, x \geq 0, b$ ganzz., A total unimodular:
 \Rightarrow Jede Basislösung ist ganzzahlig.

Bedeutet: ein ILP mit TUM Restriktionsmatrix \searrow LP!

Definition

Matrix A total unimodular falls jede quadratische Submatrix Determinante aus $\{-1, 0, 1\}$ hat.

NB: Die Elemente von A sind alle aus $\{-1, 0, 1\}$

Theorem

LP: $Ax \leq b, x \geq 0, b$ ganzz., A total unimodular:
 \Rightarrow Jede Basislösung ist ganzzahlig.

Bedeutet: ein ILP mit TUM Restriktionsmatrix \searrow LP!

Definition

Matrix A total unimodular falls jede quadratische Submatrix Determinante aus $\{-1, 0, 1\}$ hat.

NB: Die Elemente von A sind alle aus $\{-1, 0, 1\}$

Theorem

LP: $Ax \leq b, x \geq 0, b$ ganzz., A total unimodular:
 \Rightarrow Jede Basislösung ist ganzzahlig.

Bedeutet: ein ILP mit TUM Restriktionsmatrix \searrow LP!

Definition

Matrix A total unimodular falls jede quadratische Submatrix Determinante aus $\{-1, 0, 1\}$ hat.

NB: Die Elemente von A sind alle aus $\{-1, 0, 1\}$

Theorem

LP: $Ax \leq b, x \geq 0, b$ ganzz., A total unimodular:
 \Rightarrow Jede Basislösung ist ganzzahlig.

Bedeutet: ein ILP mit TUM Restriktionsmatrix \searrow LP!

Anwendungen von totaler Unimodularität

Geg.: Matching Problem im bipartiten Graph $G = (VE)$:

$$S(G) = \max\{\vec{1}^T x \mid Ax \leq \vec{1}, x \geq 0, x \text{ integer}\}$$

Lösung durch ein ILP ?

Anwendungen von totaler Unimodularität

Geg.: Matching Problem im bipartiten Graph $G = (VE)$:

$$S(G) = \max\{\vec{1}^T x \mid Ax \leq \vec{1}, x \geq 0, x \text{ integer}\}$$

Lösung durch ein ILP ?

Anwendungen von totaler Unimodularität

Theorem

Inzidenzmatrix A TUM \iff Graph bipartit

Beweis.

\Leftarrow : Ang.: G nicht bipartit \Rightarrow , \exists ungerader Kreis K

\Rightarrow Submatrix von A bzgl. K hat Det. 2

\Rightarrow Widerspruch

\Rightarrow : Sei G bipartit

\Rightarrow per Induktion über $t \times t$ Submatrix



Anwendungen von totaler Unimodularität

Theorem

Inzidenzmatrix A TUM \iff Graph bipartit

Beweis.

\Leftarrow : Ang.: G nicht bipartit \Rightarrow , \exists ungerader Kreis K

\Rightarrow Submatrix von A bzgl. K hat Det. 2

\Rightarrow Widerspruch

\Rightarrow : Sei G bipartit

\Rightarrow per Induktion über $t \times t$ Submatrix



Anwendungen von totaler Unimodularität

Theorem

Inzidenzmatrix A TUM \iff Graph bipartit

Beweis.

\Leftarrow : Ang.: G nicht bipartit \Rightarrow , \exists ungerader Kreis K

\Rightarrow Submatrix von A bzgl. K hat Det. 2

\Rightarrow Widerspruch

\Rightarrow : Sei G bipartit

\Rightarrow per Induktion über $t \times t$ Submatrix □

Anwendungen von totaler Unimodularität

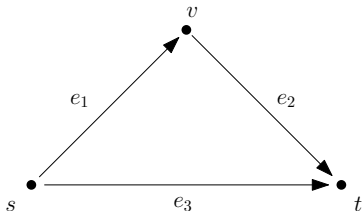
- Somit ist das matching Problem als LP lösbar
- Auch Inzidenzmatritzen von gerichteten Graphen sind TUM
- Intervall-Matritzen sind TUM
- ...

Anwendungen von totaler Unimodularität

- Somit ist das matching Problem als LP lösbar
- Auch Inzidenzmatritzen von gerichteten Graphen sind TUM
- Intervall-Matritzen sind TUM
- ...

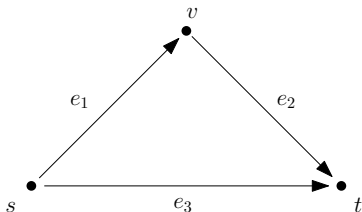
Anwendungen von totaler Unimodularität

- Somit ist das matching Problem als LP lösbar
- Auch Inzidenzmatritzen von gerichteten Graphen sind TUM
- Intervall-Matritzen sind TUM
- ...



Kantenkapazitäten $c(e_1) := 1$, $c(e_2) := 2$ und $c(e_3) := 3$.

Stellen Sie das Lineare Programm des maximalen Flussproblems für dieses Netzwerk in der in der Vorlesung gegebenen Form auf und bringen Sie es dann in die ebenfalls in der Vorlesung definierte Standardform. Stellen Sie anschließend das zur Standardform duale lineare Programm auf.



Kantenkapazitäten $c(e_1) := 1$, $c(e_2) := 2$ und $c(e_3) := 3$.

Stellen Sie das Lineare Programm des maximalen Flussproblems für dieses Netzwerk in der in der Vorlesung gegebenen Form auf und bringen Sie es dann in die ebenfalls in der Vorlesung definierte Standardform. Stellen Sie anschließend das zur Standardform duale lineare Programm auf.

Kantenkapazitätsbedingungen:

$$e_1 \leq 1$$

$$e_2 \leq 2$$

$$e_3 \leq 3$$

Flusserhaltungsbedingung für den Knoten v :

$$e_1 - e_2 = 0$$

Nichtnegativität der Kantenflüsse:

$$e_1 \geq 0$$

$$e_2 \geq 0$$

$$e_3 \geq 0$$

Kantenkapazitätsbedingungen:

$$e_1 \leq 1$$

$$e_2 \leq 2$$

$$e_3 \leq 3$$

Flusserhaltungsbedingung für den Knoten v :

$$e_1 - e_2 = 0$$

Nichtnegativität der Kantenflüsse:

$$e_1 \geq 0$$

$$e_2 \geq 0$$

$$e_3 \geq 0$$

Kantenkapazitätsbedingungen:

$$e_1 \leq 1$$

$$e_2 \leq 2$$

$$e_3 \leq 3$$

Flusserhaltungsbedingung für den Knoten v :

$$e_1 - e_2 = 0$$

Nichtnegativität der Kantenflüsse:

$$e_1 \geq 0$$

$$e_2 \geq 0$$

$$e_3 \geq 0$$

Kantenkapazitätsbedingungen:

$$e_1 \leq 1$$

$$e_2 \leq 2$$

$$e_3 \leq 3$$

Flusserhaltungsbedingung für den Knoten v :

$$e_1 - e_2 = 0$$

Nichtnegativität der Kantenflüsse:

$$e_1 \geq 0$$

$$e_2 \geq 0$$

$$e_3 \geq 0$$

Die Bestimmung des maximalen Flusses entspricht in der Standardform aus der Vorlesung dem Minimierungsproblem

$$\min(-e_1 - e_3)$$

mit folgenden Nebenbedingungen:

$$e_1 - e_2 \geq 0 \quad e_1 \geq 0$$

$$-e_1 + e_2 \geq 0 \quad e_2 \geq 0$$

$$-e_1 \geq -1 \quad e_3 \geq 0$$

$$-e_2 \geq -2$$

$$-e_3 \geq -3$$

Die Bestimmung des maximalen Flusses entspricht in der Standardform aus der Vorlesung dem Minimierungsproblem

$$\min(-e_1 - e_3)$$

mit folgenden Nebenbedingungen:

$$e_1 - e_2 \geq 0 \quad e_1 \geq 0$$

$$-e_1 + e_2 \geq 0 \quad e_2 \geq 0$$

$$-e_1 \geq -1 \quad e_3 \geq 0$$

$$-e_2 \geq -2$$

$$-e_3 \geq -3$$

(PP)

$$f(x) = x^T p = \max!$$

$$Ax \geq b$$

$$x \geq 0$$

$$M = \{Ax \leq b, x \geq 0\}$$

(DP)

$$g(u) = u^T b = \min!$$

$$A^T u \leq p$$

$$u \geq 0$$

$$N = \{A^T u \leq p, u \geq 0\}$$

Das primale Programm ist:

$$P: \quad \min c^T x \quad \text{unter} \\ Ax \geq b \quad \text{und} \quad x \geq 0$$

wobei gilt:

$$x \in \mathbb{R}^3, c = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} \in \mathbb{R}^3, b = \begin{pmatrix} 0 \\ 0 \\ -1 \\ -2 \\ -3 \end{pmatrix} \in \mathbb{R}^5, A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \in \mathbb{R}^{5 \times 3}$$

Das primale Programm ist:

$$P: \quad \min c^T x \quad \text{unter} \\ Ax \geq b \quad \text{und} \quad x \geq 0$$

wobei gilt:

$$x \in \mathbb{R}^3, c = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} \in \mathbb{R}^3, b = \begin{pmatrix} 0 \\ 0 \\ -1 \\ -2 \\ -3 \end{pmatrix} \in \mathbb{R}^5, A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \in \mathbb{R}^{5 \times 3}$$

Das zugehörige duale Programm ist:

$$D: \quad \max y^T b \quad \text{unter} \\ y^T A \leq c^T \quad \text{und} \quad y \geq 0$$

wobei gilt:

$$y \in \mathbb{R}^5, b = \begin{pmatrix} 0 \\ 0 \\ -1 \\ -2 \\ -3 \end{pmatrix} \in \mathbb{R}^5, c = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} \in \mathbb{R}^3, A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \in \mathbb{R}^{5 \times 3}$$

Das zugehörige duale Programm ist:

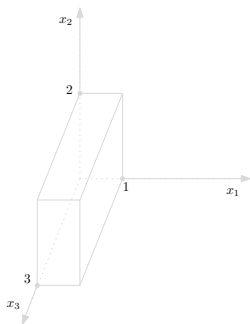
$$D: \quad \max y^T b \quad \text{unter} \\ y^T A \leq c^T \quad \text{und} \quad y \geq 0$$

wobei gilt:

$$y \in \mathbb{R}^5, b = \begin{pmatrix} 0 \\ 0 \\ -1 \\ -2 \\ -3 \end{pmatrix} \in \mathbb{R}^5, c = \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} \in \mathbb{R}^3, A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \in \mathbb{R}^{5 \times 3}$$

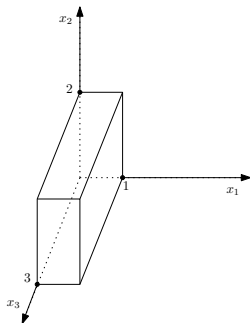
Ist das Lösungspolyeder beschränkt?

Ja, das Lösungspolyeder ist durch den Hyperquader beschränkt, der durch die Kapazitäts- und Positivitätsbedingungen bestimmt wird.



Ist das Lösungspolyeder beschränkt?

Ja, das Lösungspolyeder ist durch den Hyperquader beschränkt, der durch die Kapazitäts- und Positivitätsbedingungen bestimmt wird.



LP für den Spielzeugfluss

Stellen sie das durch die Kapazitätsbedingungen gegebene konvexe Polyeder graphisch dar, sowie die durch die Flussenerhaltungsbedingungen gegebene Hyperebene.

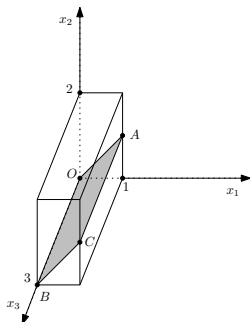


Abbildung: Graphische Darstellung des Lösungsraums in grau.

Führen Sie die Simplexmethode auf dem *Lösungs*-Polyeder durch!

- Verbessernden Kanten?

- $(0, A), (0, B)$, wobei $A = (1, 1, 0), B = (0, 0, 3)$

- Welchen Flusserhöhungen entsprechen diese?

- $(0, A) : f(1)+ = 1, f(2)+ = 1, f(3)+ = 0$, und

- $(0, B) : f(1)+ = 0, f(2)+ = 0, f(3)+ = 3$

- Wie heißt der Extrempunkt, der dem maximalen Fluss entspricht?

- $C = (1, 1, 3)$

Führen Sie die Simplexmethode auf dem *Lösungs*-Polyeder durch!

- Verbessernden Kanten?
- $(0, A), (0, B)$, wobei $A = (1, 1, 0), B = (0, 0, 3)$
- Welchen Flussserhöhungen entsprechen diese?
 - $(0, A) : f(1)+ = 1, f(2)+ = 1, f(3)+ = 0$, und
 $(0, B) : f(1)+ = 0, f(2)+ = 0, f(3)+ = 3$
- Wie heißt der Extrempunkt, der dem maximalen Fluss entspricht?
 - $C = (1, 1, 3)$

Führen Sie die Simplexmethode auf dem *Lösungs*-Polyeder durch!

- Verbessernden Kanten?
 - $(0, A), (0, B)$, wobei $A = (1, 1, 0), B = (0, 0, 3)$
- Welchen Flussserhöhungen entsprechen diese?
 - $(0, A) : f(1)+ = 1, f(2)+ = 1, f(3)+ = 0$, und
 $(0, B) : f(1)+ = 0, f(2)+ = 0, f(3)+ = 3$
- Wie heißt der Extrempunkt, der dem maximalen Fluss entspricht?
 - $C = (1, 1, 3)$

Führen Sie die Simplexmethode auf dem *Lösungs*-Polyeder durch!

- Verbessernden Kanten?
- $(0, A), (0, B)$, wobei $A = (1, 1, 0), B = (0, 0, 3)$
- Welchen Flussserhöhungen entsprechen diese?
- $(0, A) : f(1)+ = 1, f(2)+ = 1, f(3)+ = 0$, und
 $(0, B) : f(1)+ = 0, f(2)+ = 0, f(3)+ = 3$
- Wie heißt der Extrempunkt, der dem maximalen Fluss entspricht?
- $C = (1, 1, 3)$

Betrachten wir nun ein allgemeines Flussproblem.

$$\max \sum_{(s,i) \in E} x_{s,i} - \sum_{(i,s) \in E} x_{i,s}$$

unter den Nebenbedingungen

$$\left. \begin{array}{l} x_{i,j} \leq c_{i,j} \\ x_{i,j} \geq 0 \end{array} \right\} \quad \forall (i,j) \in E$$
$$\sum_{i:(i,j) \in E} x_{i,j} - \sum_{i:(j,i) \in E} x_{j,i} = 0 \quad \forall j \in V \setminus \{s, t\}$$

In Matrixform laute dies

$$\max a^T x$$

unter den Nebenbedingungen

$$\begin{aligned} \mathbb{I} \quad x &\leq c \\ -\mathbb{I} \quad x &\leq 0 \\ B \quad x &= 0 . \end{aligned}$$

Dabei sei \mathbb{I} eine Einheitsmatrix und B die Matrix, die sich aus den Flusserhaltungsbedingungen ergibt.

Welche Dimensionen haben a , I und B ?

Sei $m = |E|$ und $n = |V|$. Dann gilt:

- $a \in \mathbb{R}^m$: Ein Fluss entspricht der Belegung aller Kanten $e \in E$ mit den jeweiligen Flusswerten $f(e)$.
- $I \in \mathbb{R}^{m \times m}$: Für jede Kante gibt es eine Kapazitätsbedingung.
- $B \in \mathbb{R}^{n-2 \times m}$: Für jeden der $n - 2$ Knoten aus $V \setminus \{s, t\}$ wird die Flusserhaltungsbedingung als Gleichung über die an ihm inzidenten Kanten ausgedrückt.

Welche Dimensionen haben a , \mathbb{I} und B ?

Sei $m = |E|$ und $n = |V|$. Dann gilt:

- $a \in \mathbb{R}^m$: Ein Fluss entspricht der Belegung aller Kanten $e \in E$ mit den jeweiligen Flusswerten $f(e)$.
- $\mathbb{I} \in \mathbb{R}^{m \times m}$: Für jede Kante gibt es eine Kapazitätsbedingung.
- $B \in \mathbb{R}^{n-2 \times m}$: Für jeden der $n - 2$ Knoten aus $V \setminus \{s, t\}$ wird die Flusserhaltungsbedingung als Gleichung über die an ihm inzidenten Kanten ausgedrückt.

Welche Dimensionen haben a , \mathbb{I} und B ?

Sei $m = |E|$ und $n = |V|$. Dann gilt:

- $a \in \mathbb{R}^m$: Ein Fluss entspricht der Belegung aller Kanten $e \in E$ mit den jeweiligen Flusswerten $f(e)$.
- $\mathbb{I} \in \mathbb{R}^{m \times m}$: Für jede Kante gibt es eine Kapazitätsbedingung.
- $B \in \mathbb{R}^{n-2 \times m}$: Für jeden der $n - 2$ Knoten aus $V \setminus \{s, t\}$ wird die Flusserhaltungsbedingung als Gleichung über die an ihm inzidenten Kanten ausgedrückt.

Welche Dimensionen haben a , I und B ?

Sei $m = |E|$ und $n = |V|$. Dann gilt:

- $a \in \mathbb{R}^m$: Ein Fluss entspricht der Belegung aller Kanten $e \in E$ mit den jeweiligen Flusswerten $f(e)$.
- $I \in \mathbb{R}^{m \times m}$: Für jede Kante gibt es eine Kapazitätsbedingung.
- $B \in \mathbb{R}^{n-2 \times m}$: Für jeden der $n - 2$ Knoten aus $V \setminus \{s, t\}$ wird die Flusserhaltungsbedingung als Gleichung über die an ihm inzidenten Kanten ausgedrückt.

Welche Dimensionen haben a , \mathbb{I} und B ?

Sei $m = |E|$ und $n = |V|$. Dann gilt:

- $a \in \mathbb{R}^m$: Ein Fluss entspricht der Belegung aller Kanten $e \in E$ mit den jeweiligen Flusswerten $f(e)$.
- $\mathbb{I} \in \mathbb{R}^{m \times m}$: Für jede Kante gibt es eine Kapazitätsbedingung.
- $B \in \mathbb{R}^{n-2 \times m}$: Für jeden der $n - 2$ Knoten aus $V \setminus \{s, t\}$ wird die Flusserhaltungsbedingung als Gleichung über die an ihm inzidenten Kanten ausgedrückt.

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_j z_j = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_j z_j = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow L.U.

Allgemeines Fluss-LP

Zeigen oder widerlegen Sie: Die Zeilen der Matrix B sind linear unabhängig.

- Unzusammenhängend? Sei Flussgraph zusammenhängend
- Sei $LK = \sum \alpha_i z_i = 0$ eine Linearkombination der Zeilen
- $\alpha_v \neq 0 \Rightarrow v \in S$, sonst $v \in V \setminus S \Rightarrow$ Schnitt
- Zsh. $\Rightarrow \exists (u, v)$ oder $(v, u) \in E$ mit $u \in S, v \in V \setminus S$
- $u \in S \Rightarrow \alpha_u \neq 0$ und $v \in V \setminus S \Rightarrow \alpha_v = 0$.
- Betrachte im Ergebnis der LK diejenige Koordinate, die zur Kante (u, v) gehört: $LK_{(u,v)} \neq 0$
- $\Rightarrow \alpha_v = 0 \quad \forall v$
- Nur triviale Linearkombination $LK = \sum \alpha_i z_i = 0$ möglich
- \Rightarrow **L.U.**

Welche Dimension hat im Allgemeinen das durch die Nebenbedingungen definierte Lösungspolyeder?

- Zeile von $B \iff$ Hyperebene der Dimension $m - 1$ im \mathbb{R}^m .
- $c_i \geq 0 \Rightarrow$ Kapazitätsbedingungen-Hyperquader nichtleer (O);
- Hyperebene \cap Hyperquader \cap andere Hyperebenen $\neq \emptyset$ (O).
- $\exists n - 2$ L.U. Hyperebenen.
- \Rightarrow Dimension des Lösungspolyeder: $m - (n - 2)$.

Welche Dimension hat im Allgemeinen das durch die Nebenbedingungen definierte Lösungspolyeder?

- Zeile von $B \iff$ Hyperebene der Dimension $m - 1$ im \mathbb{R}^m .
- $c_i \geq 0 \Rightarrow$ Kapazitätsbedingungen-Hyperquader nichtleer (O);
- Hyperebene \cap Hyperquader \cap andere Hyperebenen $\neq \emptyset$ (O).
- $\exists n - 2$ L.U. Hyperebenen.
- \Rightarrow Dimension des Lösungspolyeder: $m - (n - 2)$.

Welche Dimension hat im Allgemeinen das durch die Nebenbedingungen definierte Lösungspolyeder?

- Zeile von $B \iff$ Hyperebene der Dimension $m - 1$ im \mathbb{R}^m .
- $c_i \geq 0 \Rightarrow$ Kapazitätsbedingungen-Hyperquader nichtleer (O);
- Hyperebene \cap Hyperquader \cap andere Hyperebenen $\neq \emptyset$ (O).
- $\exists n - 2$ L.U. Hyperebenen.
- \Rightarrow Dimension des Lösungspolyeder: $m - (n - 2)$.

Welche Dimension hat im Allgemeinen das durch die Nebenbedingungen definierte Lösungspolyeder?

- Zeile von $B \iff$ Hyperebene der Dimension $m - 1$ im \mathbb{R}^m .
- $c_i \geq 0 \Rightarrow$ Kapazitätsbedingungen-Hyperquader nichtleer (O);
- Hyperebene \cap Hyperquader \cap andere Hyperebenen $\neq \emptyset$ (O).
- $\exists n - 2$ L.U. Hyperebenen.
- \Rightarrow Dimension des Lösungspolyeder: $m - (n - 2)$.

Welche Dimension hat im Allgemeinen das durch die Nebenbedingungen definierte Lösungspolyeder?

- Zeile von $B \iff$ Hyperebene der Dimension $m - 1$ im \mathbb{R}^m .
- $c_i \geq 0 \Rightarrow$ Kapazitätsbedingungen-Hyperquader nichtleer (O);
- Hyperebene \cap Hyperquader \cap andere Hyperebenen $\neq \emptyset$ (O).
- $\exists n - 2$ L.U. Hyperebenen.
- \Rightarrow Dimension des Lösungspolyeder: $m - (n - 2)$.

Welche Dimension hat im Allgemeinen das durch die Nebenbedingungen definierte Lösungspolyeder?

- Zeile von $B \iff$ Hyperebene der Dimension $m - 1$ im \mathbb{R}^m .
- $c_i \geq 0 \Rightarrow$ Kapazitätsbedingungen-Hyperquader nichtleer (O);
- Hyperebene \cap Hyperquader \cap andere Hyperebenen $\neq \emptyset$ (O).
- $\exists n - 2$ L.U. Hyperebenen.
- \Rightarrow Dimension des Lösungspolyeder: $m - (n - 2)$.

Zeigen oder widerlegen Sie: Die Erhöhung des Flusses entlang eines erhöhenden Weges entspricht einem Schritt im Simplexverfahren.

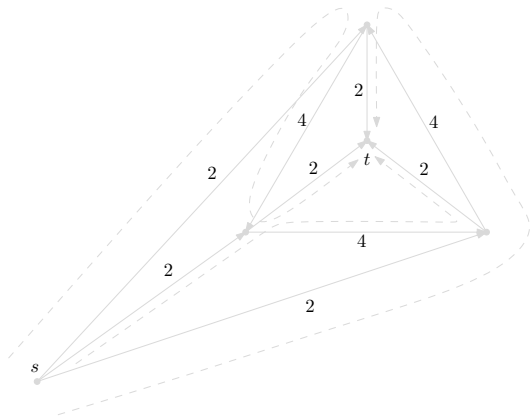


Abbildung: Gegenbeispiel für Teilaufgabe (h).

Zeigen oder widerlegen Sie: Die Erhöhung des Flusses entlang eines erhöhenden Weges entspricht einem Schritt im Simplexverfahren.

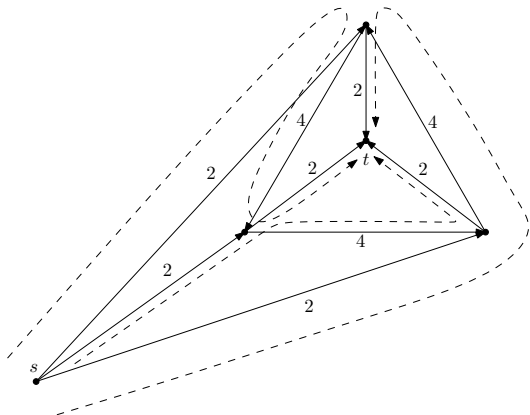


Abbildung: Gegenbeispiel für Teilaufgabe (h).

- Erhöhe Fluss folgendermaßen
- $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
- Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
- $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
- \Rightarrow kein Simplexschritt

- Erhöhe Fluss folgendermaßen
 - $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
 - $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
 - $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
 - Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
 - $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
 - \Rightarrow kein Simplexschritt

- Erhöhe Fluss folgendermaßen
- $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
- Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
- $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
- \Rightarrow kein Simplexschritt

- Erhöhe Fluss folgendermaßen
- $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
- Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
- $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
- \Rightarrow kein Simplexschritt

- Erhöhe Fluss folgendermaßen
- $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
- Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
- $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
- \Rightarrow kein Simplexschritt

- Erhöhe Fluss folgendermaßen
- $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
- Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
- $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
- \Rightarrow kein Simplexschritt

- Erhöhe Fluss folgendermaßen
- $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
- Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
- $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
- \Rightarrow kein Simplexschritt

- Erhöhe Fluss folgendermaßen
- $(2, 2, 2, 0, 0, 0, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 0, 0, 0)$
- $(2, 2, 2, 2, 2, 2, 2, 2, 2)$
- Aber: $(2, 2, 2, 2, 2, 2, 2, 2, 2) =$
 $1/2 \cdot (2, 2, 2, 4, 4, 4, 2, 2, 2) + 1/2 \cdot (2, 2, 2, 0, 0, 0, 2, 2, 2)$
wobei $(2, 2, 2, 4, 4, 4, 2, 2, 2)$, $(2, 2, 2, 0, 0, 0, 2, 2, 2)$ zulässig!
- $\Rightarrow (2, 2, 2, 2, 2, 2, 2, 2, 2)$ keine Ecke
- \Rightarrow kein Simplexschritt

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flusserhaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flusserhaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flusserhaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flusserhaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flusserhaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flusserhaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flussershaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flussershaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flusserhaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flusserhaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flusserhaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flusserhaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flusserhaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flusserhaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Zeigen oder widerlegen Sie die Umkehrung des vorherigen Satzes.

- Simplexschritt: Wechsel von Ecke zu Ecke im Lösungspolyeder, in Richtung der Zielfunktion.
- Zielfunktion = Größe des s - t -Flusses.

Fall I: Simplexschritt erhöht s - t -Fluss

- \exists eine Kante (s, v) , wo Fluss erhöht wurde.
- Flusserhaltung bei $v \Rightarrow \exists$ Kante (v, w) wo Fluss erhöht wurde.
- Einziger Knoten an dem Kette enden kann ist t (dort gibt es keine Flusserhaltung).
- (s kommt als Ende nicht in Frage, sonst keine s - t -Flusserhöhung, sondern Kreis)
- \Rightarrow Erhöhender s - t -Weg (der nicht notwendigerweise einfach sein muss) gefunden.

Fall II: Simplexschritt erhöht s - t -Fluss nicht

- Lösungspolyeder enthält viele degenerierte Ecken.
 - \exists Ecken deren Nachbarn nicht zu einer Verbesserung führen.
 - ↑ Solche Simplex-Schritte dienen der “Breitensuche” nach erhöhendem s - t -Pfad.
- ⇒ Simplex-Schritt entspricht nicht erhöhendem Weg.

Sobald Breitensuche vollständigen s - t -Pfad gefunden hat, tritt Fall I ein.

Fall II: Simplexschritt erhöht s - t -Fluss nicht

- Lösungspolyeder enthält viele degenerierte Ecken.
 - \exists Ecken deren Nachbarn nicht zu einer Verbesserung führen.
 - \uparrow Solche Simplex-Schritte dienen der "Breitensuche" nach erhöhendem s - t -Pfad.
- \Rightarrow Simplex-Schritt entspricht nicht erhöhendem Weg.

Sobald Breitensuche vollständigen s - t -Pfad gefunden hat, tritt Fall I ein.

Fall II: Simplexschritt erhöht s - t -Fluss nicht

- Lösungspolyeder enthält viele degenerierte Ecken.
- \exists Ecken deren Nachbarn nicht zu einer Verbesserung führen.

↑ Solche Simplex-Schritte dienen der "Breitensuche" nach erhöhendem s - t -Pfad.

⇒ Simplex-Schritt entspricht nicht erhöhendem Weg.

Sobald Breitensuche vollständigen s - t -Pfad gefunden hat, tritt Fall I ein.

Fall II: Simplexschritt erhöht s - t -Fluss nicht

- Lösungspolyeder enthält viele degenerierte Ecken.
- \exists Ecken deren Nachbarn nicht zu einer Verbesserung führen.

○
↑ Solche Simplex-Schritte dienen der “Breitensuche” nach erhöhendem s - t -Pfad.

⇒ Simplex-Schritt entspricht nicht erhöhendem Weg.

Sobald Breitensuche vollständigen s - t -Pfad gefunden hat, tritt Fall I ein.

Fall II: Simplexschritt erhöht s - t -Fluss nicht

- Lösungspolyeder enthält viele degenerierte Ecken.
- \exists Ecken deren Nachbarn nicht zu einer Verbesserung führen.

○
↑ Solche Simplex-Schritte dienen der “Breitensuche” nach erhöhendem s - t -Pfad.

⇒ Simplex-Schritt entspricht nicht erhöhendem Weg.

Sobald Breitensuche vollständigen s - t -Pfad gefunden hat, tritt Fall I ein.

Fall II: Simplexschritt erhöht s - t -Fluss nicht

- Lösungspolyeder enthält viele degenerierte Ecken.
- \exists Ecken deren Nachbarn nicht zu einer Verbesserung führen.

○
↑ Solche Simplex-Schritte dienen der “Breitensuche” nach erhöhendem s - t -Pfad.

⇒ Simplex-Schritt entspricht nicht erhöhendem Weg.

Sobald Breitensuche vollständigen s - t -Pfad gefunden hat, tritt Fall I ein.

**Danke für Eure
Aufmerksamkeit!**

**Danke für Eure
Aufmerksamkeit!**

Fragen?