Algorithmentechnik — Übung 6

http://i11www.ira.uka.de/teaching/WS_0506/algotech

Robert Görke (rgoerke@ira.uka.de)

WS 05/06

Übersicht

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

- Z.z.: $\forall v \in (c_1 \oplus c_2) : \deg(v)$ ist gerade

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

- Z.z.: $\forall v \in (c_1 \oplus c_2) : \deg(v)$ ist gerade
- $c_1 \oplus c_2 = (c_1 \cup c_2) \setminus (c_1 \cap c_2)$

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

- Z.z.: $\forall v \in (c_1 \oplus c_2) : \deg(v)$ ist gerade
- $c_1 \oplus c_2 = (c_1 \cup c_2) \setminus (c_1 \cap c_2)$

1. $v \notin c_1, c_2 \Rightarrow v \notin c_1 \oplus c_2$

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

- Z.z.: $\forall v \in (c_1 \oplus c_2) : \deg(v)$ ist gerade
- $c_1 \oplus c_2 = (c_1 \cup c_2) \setminus (c_1 \cap c_2)$

1. $v \notin c_1, c_2 \Rightarrow v \notin c_1 \oplus c_2$
2. $v \in c_1, v \notin c_2 \Rightarrow \deg(v)$ ist gerade in $c_1 \oplus c_2$

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

- Z.z.: $\forall v \in (c_1 \oplus c_2) : \deg(v)$ ist gerade
- $c_1 \oplus c_2 = (c_1 \cup c_2) \setminus (c_1 \cap c_2)$

1. $v \notin c_1, c_2 \Rightarrow v \notin c_1 \oplus c_2$
2. $v \in c_1, v \notin c_2 \Rightarrow \deg(v)$ ist gerade in $c_1 \oplus c_2$
3. $v \in c_2, v \notin c_1 \Rightarrow \deg(v)$ ist gerade in $c_1 \oplus c_2$

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

- Z.z.: $\forall v \in (c_1 \oplus c_2) : \deg(v)$ ist gerade
- $c_1 \oplus c_2 = (c_1 \cup c_2) \setminus (c_1 \cap c_2)$

1. $v \notin c_1, c_2 \Rightarrow v \notin c_1 \oplus c_2$
2. $v \in c_1, v \notin c_2 \Rightarrow \deg(v)$ ist gerade in $c_1 \oplus c_2$
3. $v \in c_2, v \notin c_1 \Rightarrow \deg(v)$ ist gerade in $c_1 \oplus c_2$
4. $v \in c_1, c_2 \Rightarrow \deg(v)_{c_1}, \deg(v)_{c_2}$ gerade. Sei $k = |\{e \in c_1 \cap c_2 | e \text{ inzident mit } v\}|$

Summe zweier Kreise

Zeige: Kreise c_1 und $c_2 \Rightarrow c_1 \oplus c_2$ wieder Kreis
(c Kreis $\Leftrightarrow \forall v \in c : \deg(v)$ ist gerade)

- Z.z.: $\forall v \in (c_1 \oplus c_2) : \deg(v)$ ist gerade
- $c_1 \oplus c_2 = (c_1 \cup c_2) \setminus (c_1 \cap c_2)$

1. $v \notin c_1, c_2 \Rightarrow v \notin c_1 \oplus c_2$
2. $v \in c_1, v \notin c_2 \Rightarrow \deg(v)$ ist gerade in $c_1 \oplus c_2$
3. $v \in c_2, v \notin c_1 \Rightarrow \deg(v)$ ist gerade in $c_1 \oplus c_2$
4. $v \in c_1, c_2 \Rightarrow \deg(v)_{c_1}, \deg(v)_{c_2}$ gerade. Sei
   $k = |\{e \in c_1 \cap c_2 | e \text{ inzident mit } v\}|$
   $\deg(v) = (\deg(v)_{c_1} + \deg(v)_{c_2} - k) - (k)$ ist gerade

Mengen von l.u. Kreisen sind Matroid

Mengen von l.u. Kreisen sind Matroid

- $\emptyset$ is l.u.

Mengen von l.u. Kreisen sind Matroid

- $\emptyset$ is l.u.
- Teilmengen linear unabhängiger Mengen sind l.u.

Mengen von l.u. Kreisen sind Matroid

- $\emptyset$ is l.u.
- Teilmengen linear unabhängiger Mengen sind l.u.
- C_1, C_2 l.u, mit $|C_1| < |C_2| \Rightarrow dim(span(C_1)) < dim(span(C_2))$

Mengen von l.u. Kreisen sind Matroid

- $\emptyset$ is l.u.
- Teilmengen linear unabhängiger Mengen sind l.u.
- C_1, C_2 l.u, mit $|C_1| < |C_2| \Rightarrow dim(span(C_1)) < dim(span(C_2))$
  $\Rightarrow \exists v \in C_2$ l.u. von C_1 (sonst $span(C_2) \subseteq span(C_1)$)

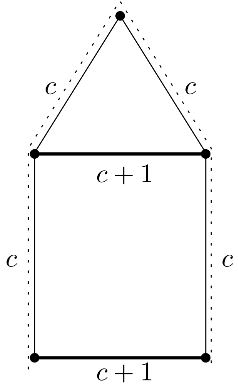
Mengen von l.u. Kreisen sind Matroid

- $\emptyset$ is l.u.
- Teilmengen linear unabhängiger Mengen sind l.u.
- C_1, C_2 l.u, mit $|C_1| < |C_2| \Rightarrow \mathit{dim}(\mathit{span}(C_1)) < \mathit{dim}(\mathit{span}(C_2))$
  $\Rightarrow \exists v \in C_2$ l.u. von C_1 (sonst $\mathit{span}(C_2) \subseteq \mathit{span}(C_1)$)
  $\Rightarrow$ Also bildet $C_1 \cup \{v\}$ die gesuchte lineare unabhängige Menge

Vier Vermutungen

Vier Vermutungen

Die Fundamentalbasis zu einem MST ist eine MCB?

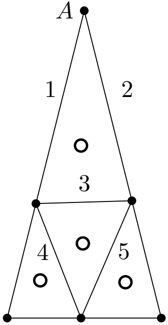


Gegenbeispiel zu (a)

Vier Vermutungen

Jede MCB ist Fundamentalbasis?

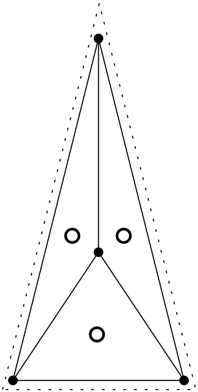
In jedem Graphen gibt es eine MCB, die Fundamentalbasis ist?



Gegenbeispiel zu (b) und (c)

Vier Vermutungen

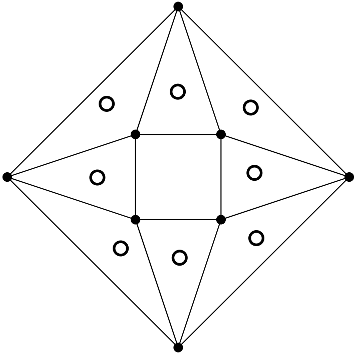
$$\mathcal{K} := \{ C_{\min}(e_i) \mid C_{\min}(e_i) \text{ kürzester Kreis, der } e_i \text{ enthält, } e_i \in E \}$$



Gegenbeispiel zu (d): keine Kreisbasis

Neue Hortonmenge?

$\mathcal{K} := \{C_{\min}(e_i) \mid C_{\min}(e_i) \text{ kürzester Kreis, der } e_i \text{ enthält, } e_i \in E\}$



Gegenbeispiel: $|\mathcal{K}| = 8$ aber
dim(MCB)$= m - n + 1 = 16 - 8 + 1 = 9$

De Pina

Ursprünglich:

1. Eingabe Graph $G = (V, E)$
2. Ausgabe MCB von G
3. Fuer $j = 1$ bis N
   Initialisiere $S_{1,j} \leftarrow \{e_i\}$
4. Fuer $k = 1$ bis N
5. Finde einen kürzesten Kreis C_k, mit $S_{k,k} \cap C_k$ ungerade
6. Fuer $j = k + 1$ bis N
7. $S_{k+1,j} \leftarrow$
8. $\begin{cases} S_{k,j} & , C_k \cap S_{k,j} \text{ gerade} \\ S_{k,j} \oplus S_{k,k} & , C_k \cap S_{k,j} \text{ ungerade} \end{cases}$

De Pina

Algebraisch:

1. Eingabe Graph $G = (V, E)$
2. Ausgabe MCB von G
3. Fuer $i = 1$ bis N
   $S_i \leftarrow \{e_i\}$
4. Fuer $k = 1$ bis N
5. Finde einen kürzesten Kreis C_k mit $\langle C_k, S_k \rangle = 1$
6. Fuer $i = k + 1$ bis N
7. Wenn $\langle C_k, S_i \rangle = 1$ dann $S_i \leftarrow S_i \oplus S_k$

De Pina

Allgemeiner (Simple MCB):

1. Eingabe Graph $G = (V, E)$
2. Ausgabe MCB von G
3. $S_1 \leftarrow \{e_1\}$
4. $C_1 \leftarrow$ kürzesten Kreis mit $\langle C_1, S_1 \rangle = 1$
5. Fuer $k = 2$ bis N
6. Berechne beliebigen Vektor S_k, der eine nichttriviale Lösung des Systems $\langle C_i, X \rangle = 0$ für $i = 1$ bis $k - 1$ ist
7. Finde kürzesten Kreis C_k, mit $\langle C_k, S_k \rangle = 1$

Basiswandel

Beweisen sie, dass $\mathcal{B}^* := \mathcal{B} \setminus \{D_j\} \cup \{C_{i+1}\}$ wieder eine Basis ist.

Basiswandel

Beweisen sie, dass $\mathcal{B}^* := \mathcal{B} \setminus \{D_j\} \cup \{C_{i+1}\}$ wieder eine Basis ist.

- Es ist: $C_{i+1} = D_1 \oplus, \ldots, \oplus D_{j-1} \oplus D_j \oplus D_{j+1} \oplus, \ldots, \oplus D_\ell$

Basiswandel

Beweisen sie, dass $\mathcal{B}^* := \mathcal{B} \setminus \{D_j\} \cup \{C_{i+1}\}$ wieder eine Basis ist.

- Es ist: $C_{i+1} = D_1 \oplus, \dots, \oplus D_{j-1} \oplus D_j \oplus D_{j+1} \oplus, \dots, \oplus D_\ell$
- Also ist: $C_{i+1} \oplus D_j = D_1 \oplus, \dots, D_{j-1}, (D_j \oplus D_j), D_{j+1}, \dots, D_\ell$

Basiswandel

Beweisen sie, dass $\mathcal{B}^* := \mathcal{B} \setminus \{D_j\} \cup \{C_{i+1}\}$ wieder eine Basis ist.

- Es ist: $C_{i+1} = D_1 \oplus, \dots, \oplus D_{j-1} \oplus D_j \oplus D_{j+1} \oplus, \dots, \oplus D_\ell$
- Also ist: $C_{i+1} \oplus D_j = D_1 \oplus, \dots, D_{j-1}, (D_j \oplus D_j), D_{j+1}, \dots, D_\ell$
- Also ist:
  $(C_{i+1} \oplus C_{i+1}) \oplus D_j = D_1 \oplus, \dots, D_{j-1} \oplus D_{j+1} \oplus, \dots, \oplus D_\ell \oplus C_{i+1}$

Basiswandel

Beweisen sie, dass $\mathcal{B}^* := \mathcal{B} \setminus \{D_j\} \cup \{C_{i+1}\}$ wieder eine Basis ist.

- Es ist: $C_{i+1} = D_1 \oplus, \ldots, \oplus D_{j-1} \oplus D_j \oplus D_{j+1} \oplus, \ldots, \oplus D_\ell$
- Also ist: $C_{i+1} \oplus D_j = D_1 \oplus, \ldots, D_{j-1}, (D_j \oplus D_j), D_{j+1}, \ldots, D_\ell$
- Also ist: $(C_{i+1} \oplus C_{i+1}) \oplus D_j = D_1 \oplus, \ldots, D_{j-1} \oplus D_{j+1} \oplus, \ldots, \oplus D_\ell \oplus C_{i+1}$
- $= D_j = D_1 \oplus, \ldots, \oplus D_{j-1} \oplus D_{j+1} \oplus, \ldots, \oplus D_\ell \oplus C_{i+1}$

Basiswandel

Beweisen sie, dass $\mathcal{B}^* := \mathcal{B} \setminus \{D_j\} \cup \{C_{i+1}\}$ wieder eine Basis ist.

- Es ist: $C_{i+1} = D_1 \oplus, \ldots, \oplus D_{j-1} \oplus D_j \oplus D_{j+1} \oplus, \ldots, \oplus D_\ell$
- Also ist: $C_{i+1} \oplus D_j = D_1 \oplus, \ldots, D_{j-1}, (D_j \oplus D_j), D_{j+1}, \ldots, D_\ell$
- Also ist: $(C_{i+1} \oplus C_{i+1}) \oplus D_j = D_1 \oplus, \ldots, D_{j-1} \oplus D_{j+1} \oplus, \ldots, \oplus D_\ell \oplus C_{i+1}$
- $= D_j = D_1 \oplus, \ldots, \oplus D_{j-1} \oplus D_{j+1} \oplus, \ldots, \oplus D_\ell \oplus C_{i+1}$
- $\in B^*$

Aufgabenstellung

Sei (D; s; t;c) ein Netzwerk, f ein Präfluss. Ein saturierter Schnitt (bzgl. f) ist ein s-t-Schnitt $(S, V \setminus S)$, so dass gilt:

$$\forall u \in S, v \in V \setminus S \colon f(u, v) = c(u, v)$$

1. Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.
2. Folgern Sie mit Hilfe von (a), dass der Algorithmus einen maximalen Fluss berechnet, falls er terminiert.
3. Für jede Kante (v, t) gilt: Aus $\mathrm{dist}(v) > 1$ folgt $f(v, t) = c(v, t)$.
4. Für jeden Knoten v gilt: Wenn es einen (bzgl. f) erhöhenden Weg von v nach t gibt, dann ist $\mathrm{dist}(v)$ eine untere Schranke für die Länge dieses Weges.

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt
- Allgemein: (Lemma 4.22) $\nexists$ s-t-Weg in D_f (Reisdualgraph)
  - Eine Kante aus E, welche noch nicht saturiert ist, ist eine Residualkante.
  - Eine Kante aus $E' \setminus E$ welche nicht leer ist ist ebenfalls eine Residualkante.

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt
- Allgemein: (Lemma 4.22) $\nexists$ s-t-Weg in D_f (Reisdualgraph)
  - Eine Kante aus E, welche noch nicht saturiert ist, ist eine Residualkante.
  - Eine Kante aus $E' \setminus E$ welche nicht leer ist ist ebenfalls eine Residualkante.
- $\nexists$ erhöhender s-t-Weg in G

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt
- Allgemein: (Lemma 4.22) $\nexists$ s-t-Weg in D_f (Reisdualgraph)
  - Eine Kante aus E, welche noch nicht saturiert ist, ist eine Residualkante.
  - Eine Kante aus $E' \setminus E$ welche nicht leer ist ist ebenfalls eine Residualkante.
- $\nexists$ erhöhender s-t-Weg in G
- Betrachte $S = \{v \mid \exists \text{ erh. } s\text{-}v\text{-Weg}\}$

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt
- Allgemein: (Lemma 4.22) $\nexists$ s-t-Weg in D_f (Reisdualgraph)
  - Eine Kante aus E, welche noch nicht saturiert ist, ist eine Residualkante.
  - Eine Kante aus $E' \setminus E$ welche nicht leer ist ist ebenfalls eine Residualkante.
- $\nexists$ erhöhender s-t-Weg in G
- Betrachte $S = \{v \mid \exists \text{ erh. } s\text{-}v\text{-Weg}\}$
- $t \notin S \Rightarrow (S, V \setminus S)$ ist s-t-Schnitt.

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt
- Allgemein: (Lemma 4.22) $\nexists$ s-t-Weg in D_f (Reisdualgraph)
  - Eine Kante aus E, welche noch nicht saturiert ist, ist eine Residualkante.
  - Eine Kante aus $E' \setminus E$ welche nicht leer ist ist ebenfalls eine Residualkante.
- $\nexists$ erhöhender s-t-Weg in G
- Betrachte $S = \{v \mid \exists$ erh. s-v-Weg$\}$
- $t \notin S \Rightarrow (S, V \setminus S)$ ist s-t-Schnitt. Saturiert ?

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt
- Allgemein: (Lemma 4.22) $\nexists$ s-t-Weg in D_f (Reisdualgraph)
  - Eine Kante aus E, welche noch nicht saturiert ist, ist eine Residualkante.
  - Eine Kante aus $E' \setminus E$ welche nicht leer ist ist ebenfalls eine Residualkante.
- $\nexists$ erhöhender s-t-Weg in G
- Betrachte $S = \{v \mid \exists$ erh. s-v-Weg$\}$
- $t \notin S \Rightarrow (S, V \setminus S)$ ist s-t-Schnitt. Saturiert ?
- Alle Kanten (in G) von S nach $V \setminus S$ sind nicht erhöhbar

Immer ein saturierter Schnitt

Zeigen Sie: Nach jeder Push- und Relabel-Operation gibt es einen saturierten Schnitt bzgl. des aktuellen Präflusses f.

- Nach der Initialisierung gibt es einen sat. Schnitt
- Allgemein: (Lemma 4.22) $\nexists$ s-t-Weg in D_f (Reisdualgraph)
  - Eine Kante aus E, welche noch nicht saturiert ist, ist eine Residualkante.
  - Eine Kante aus $E' \setminus E$ welche nicht leer ist ist ebenfalls eine Residualkante.
- $\nexists$ erhöhender s-t-Weg in G
- Betrachte $S = \{v \mid \exists$ erh. s-v-Weg$\}$
- $t \notin S \Rightarrow (S, V \setminus S)$ ist s-t-Schnitt. Saturiert ?
- Alle Kanten (in G) von S nach $V \setminus S$ sind nicht erhöhbar Saturiert !

Am Ende ein Max-Flow

Folgern Sie mit Hilfe von (a), dass der Algorithmus einen maximalen Fluss berechnet, falls er terminiert.

Am Ende ein Max-Flow

Folgern Sie mit Hilfe von (a), dass der Algorithmus einen maximalen Fluss berechnet, falls er terminiert.

- Am Ende: Wir haben sat. Schnitt $(S, V \setminus S)$ (laut (a))

Am Ende ein Max-Flow

Folgern Sie mit Hilfe von (a), dass der Algorithmus einen maximalen Fluss berechnet, falls er terminiert.

- Am Ende: Wir haben sat. Schnitt $(S, V \setminus S)$ (laut (a))
- Am Ende: Präfluss ein *Fluss*, da kein Knoten aktiv

Am Ende ein Max-Flow

Folgern Sie mit Hilfe von (a), dass der Algorithmus einen maximalen Fluss berechnet, falls er terminiert.

- Am Ende: Wir haben sat. Schnitt $(S, V \setminus S)$ (laut (a))
- Am Ende: Präfluss ein *Fluss*, da kein Knoten aktiv
-

$$c(S, V \setminus S) = \sum_{u \in S, v \notin S} c(u,v) \underbrace{=}_{\text{sat.}} \sum_{u \in S, v \notin S} f(u,v) = w(f) \quad .$$

Am Ende ein Max-Flow

Folgern Sie mit Hilfe von (a), dass der Algorithmus einen maximalen Fluss berechnet, falls er terminiert.

- Am Ende: Wir haben sat. Schnitt $(S, V \setminus S)$ (laut (a))
- Am Ende: Präfluss ein *Fluss*, da kein Knoten aktiv
- $$c(S, V \setminus S) = \sum_{u \in S, v \notin S} c(u, v) \underbrace{=}_{\text{sat.}} \sum_{u \in S, v \notin S} f(u, v) = w(f) \quad .$$
- $w(f) = c(S, V \setminus S) \geq \text{Min-Cut} = \text{Max-Flow}$

Am Ende ein Max-Flow

Folgern Sie mit Hilfe von (a), dass der Algorithmus einen maximalen Fluss berechnet, falls er terminiert.

- Am Ende: Wir haben sat. Schnitt $(S, V \setminus S)$ (laut (a))
- Am Ende: Präfluss ein *Fluss*, da kein Knoten aktiv
-

$$c(S, V \setminus S) = \sum_{u \in S, v \notin S} c(u, v) \underbrace{=}_{\text{sat.}} \sum_{u \in S, v \notin S} f(u, v) = w(f) \ .$$

- $w(f) = c(S, V \setminus S) \geq$ Min-Cut $=$ Max-Flow
- $\Rightarrow f$ ist Max-Flow

Saturierte Zielkanten bei $\mathrm{dist} = 2$

Für jede Kante (v, t) gilt: Aus $\mathrm{dist}(v) > 1$ folgt $f(v, t) = c(v, t)$.

Saturierte Zielkanten bei $\mathrm{dist} = 2$

Für jede Kante (v, t) gilt: Aus $\mathrm{dist}(v) > 1$ folgt $f(v, t) = c(v, t)$.

- Angenommen es passiert: Relabel(v) so, dass $\mathrm{dist}(v)$ auf 2

Saturierte Zielkanten bei $\mathrm{dist} = 2$

Für jede Kante (v, t) gilt: Aus $\text{dist}(v) > 1$ folgt $f(v, t) = c(v, t)$.

- Angenommen es passiert: Relabel(v) so, dass $\mathrm{dist}(v)$ auf 2
- Betrachte Relabelbedingung: zulässig, falls v aktiv, und $\forall w$ mit $r_f(v, w) > 0$ gilt, dass $\mathrm{dist}(v) \leq \mathrm{dist}(w)$

Saturierte Zielkanten bei $\mathrm{dist} = 2$

Für jede Kante (v, t) gilt: Aus $\mathrm{dist}(v) > 1$ folgt $f(v,t) = c(v,t)$.

- Angenommen es passiert: $\mathrm{Relabel}(v)$ so, dass $\mathrm{dist}(v)$ auf 2
- Betrachte Relabelbedingung: zulässig, falls v aktiv, und $\forall w$ mit $r_f(v,w) > 0$ gilt, dass $\mathrm{dist}(v) \leq \mathrm{dist}(w)$
- Aber: $\mathrm{dist}(t) = 0 \Rightarrow r_f(v,w) \leq 0$

Saturierte Zielkanten bei $\mathrm{dist} = 2$

Für jede Kante (v, t) gilt: Aus $\mathrm{dist}(v) > 1$ folgt $f(v, t) = c(v, t)$.

- Angenommen es passiert: Relabel(v) so, dass $\mathrm{dist}(v)$ auf 2
- Betrachte Relabelbedingung: zulässig, falls v aktiv, und $\forall w$ mit $r_f(v, w) > 0$ gilt, dass $\mathrm{dist}(v) \leq \mathrm{dist}(w)$
- Aber: $\mathrm{dist}(t) = 0 \Rightarrow r_f(v, w) \leq 0$
- $\Rightarrow f(v, t) = c(v, t)$

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

- $\delta(v)$ sei die Distanz von v nach t im Residualgraphen

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

- $\delta(v)$ sei die Distanz von v nach t im Residualgraphen
  Induktion nach $\delta(v)$

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

- $\delta(v)$ sei die Distanz von v nach t im Residualgraphen
  Induktion nach $\delta(v)$
- I.A.: $\delta(v) = 0$ bedeutet $v = t$ und $\mathrm{dist}(v) = 0$

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

- $\delta(v)$ sei die Distanz von v nach t im Residualgraphen
  Induktion nach $\delta(v)$
- I.A.: $\delta(v) = 0$ bedeutet $v = t$ und $\mathrm{dist}(v) = 0$
- I:V.: Behauptung gelte $\forall w$ mit $\delta(w) \leq \delta(v) - 1$

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

- $\delta(v)$ sei die Distanz von v nach t im Residualgraphen
  Induktion nach $\delta(v)$
- I.A.: $\delta(v) = 0$ bedeutet $v = t$ und $\mathrm{dist}(v) = 0$
- I:V.: Behauptung gelte $\forall w$ mit $\delta(w) \leq \delta(v) - 1$
- I.S.: Die Kante (v, w) sei die erste Kante eines kürzesten Weges von v nach t. Dann gilt $f(v, w) < c(v, w)$.

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

- $\delta(v)$ sei die Distanz von v nach t im Residualgraphen
  Induktion nach $\delta(v)$
- I.A.: $\delta(v) = 0$ bedeutet $v = t$ und $\mathrm{dist}(v) = 0$
- I:V.: Behauptung gelte $\forall w$ mit $\delta(w) \leq \delta(v) - 1$
- I.S.: Die Kante (v, w) sei die erste Kante eines kürzesten Weges von v nach t. Dann gilt $f(v, w) < c(v, w)$.
- Also gilt auch $\mathrm{dist}(v) \leq \mathrm{dist}(w) + 1$ wegen der Zulässigkeit von dist.

Länge eines erhöhenden Weges

Für jeden Knoten v gilt: $\exists$ erhöhenden Weg von v nach $t \Rightarrow$ $\mathrm{dist}(v)$ ist eine untere Schranke für die Länge dieses Weges.

- $\delta(v)$ sei die Distanz von v nach t im Residualgraphen
  Induktion nach $\delta(v)$
- I.A.: $\delta(v) = 0$ bedeutet $v = t$ und $\mathrm{dist}(v) = 0$
- I:V.: Behauptung gelte $\forall w$ mit $\delta(w) \leq \delta(v) - 1$
- I.S.: Die Kante (v, w) sei die erste Kante eines kürzesten Weges von v nach t. Dann gilt $f(v, w) < c(v, w)$.
- Also gilt auch $\mathrm{dist}(v) \leq \mathrm{dist}(w) + 1$ wegen der Zulässigkeit von dist.
- $\Rightarrow \mathrm{dist}(v) \leq \mathrm{dist}(w) + 1 \underbrace{\leq}_{\text{I.V.}} \delta(w) + 1 = \delta(v)$.

Beispiel: Einfache Bäckersrechnung

Weizenmischbrot

- Zutaten für eine Kiste Weizenmischbrot:
  - 12kg Weizenmehl
  - 8kg Wasser
- Gewinn pro Kiste: 20 Euro

Beispiel: Einfache Bäckersrechnung

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

Beispiel: Einfache Bäckersrechnung

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 Bäckersrechnung
Schematische Darstellung

	Weizenmehl	*Wasser*	*Mischkornschrot*
Weizenmischbrot	12 kg	8 kg	0 kg
Mehrkornbrot	6 kg	12 kg	10 kg

Beispiel: Einfache Bäckersrechnung
Schematische Darstellung

	Weizenmehl	*Wasser*	*Mischkornschrot*
Weizenmischbrot	12 kg	8 kg	0 kg
Mehrkornbrot	6 kg	12 kg	10 kg
Kontingent	630 kg	620 kg	350 kg

Beispiel: Einfache Bäckersrechnung
Schematische Darstellung

	Weizenmehl	*Wasser*	*Mischkornschrot*
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!

Beispiel: Einfache Bäckersrechnung
Schematische Darstellung

	Weizenmehl	*Wasser*	*Mischkornschrot*
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 Bäckersrechnung
Mathematische Formulierung

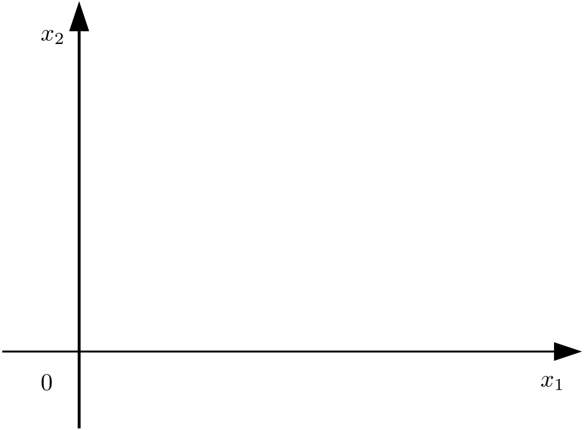
Seien $x_1 =$ Kisten Weizenmischbrot, $x_2 =$ Kisten Mehrkornbrot:

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

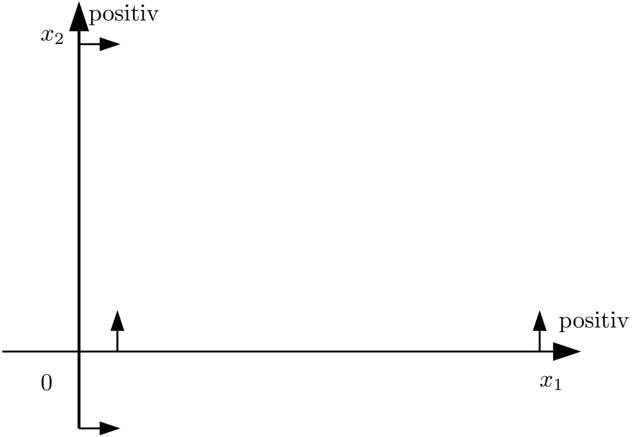
Nebenbedingungen **NB:**

$$
\begin{aligned}
12x_1 + 6\,x_2 &\leq 630 \\
8\,x_1 + 12x_2 &\leq 620 \\
10x_2 &\leq 350 \\
x_1 &\geq 10
\end{aligned}
$$

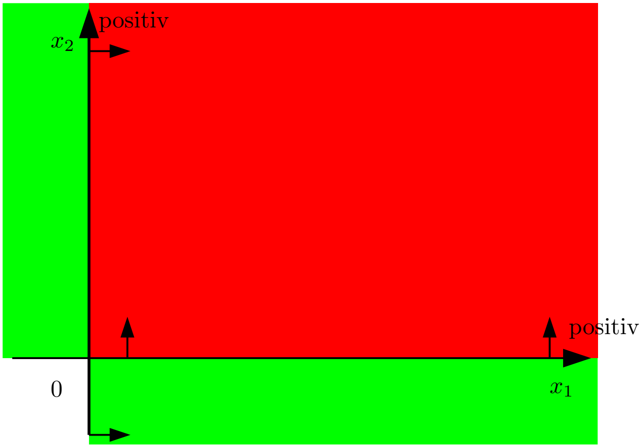
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



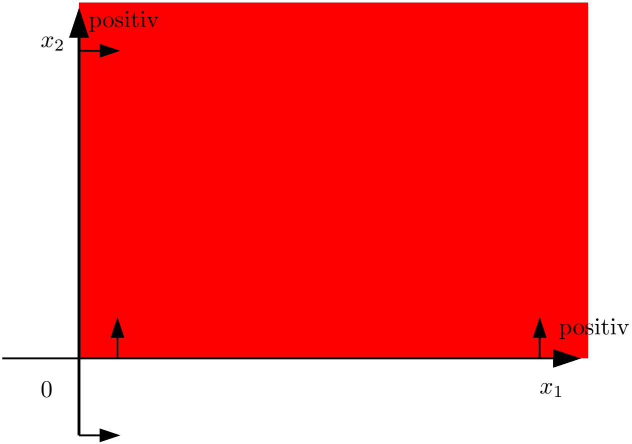
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



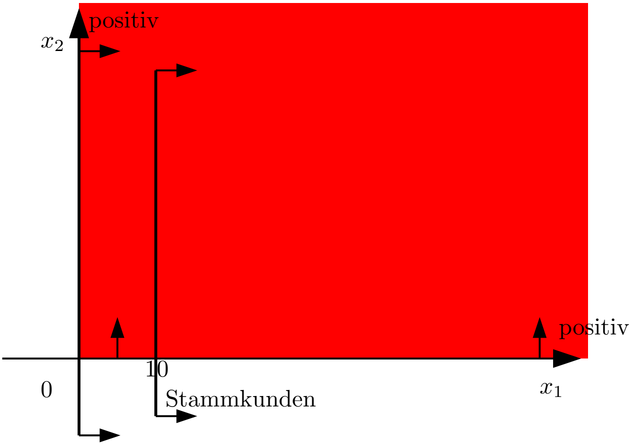
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



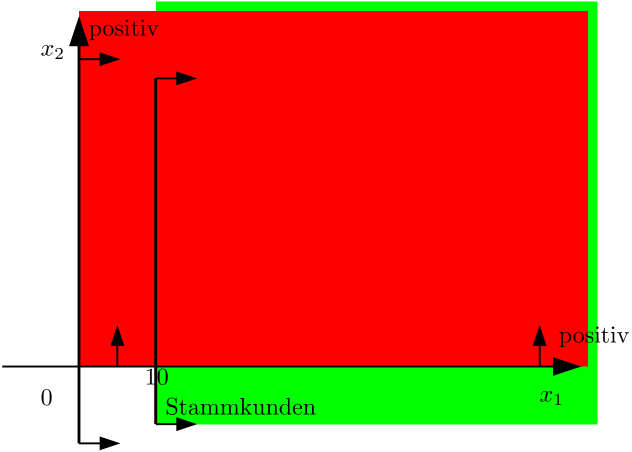
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



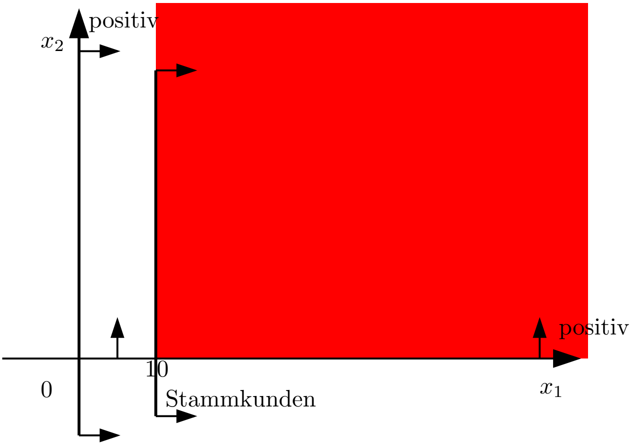
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



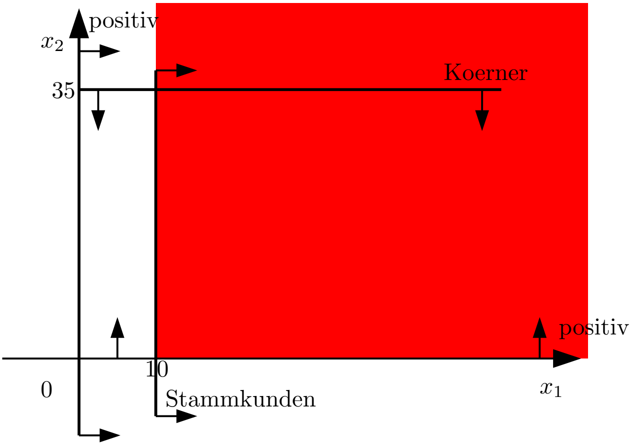
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



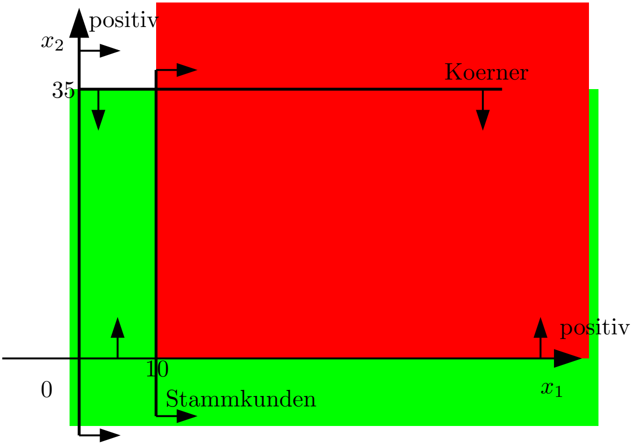
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



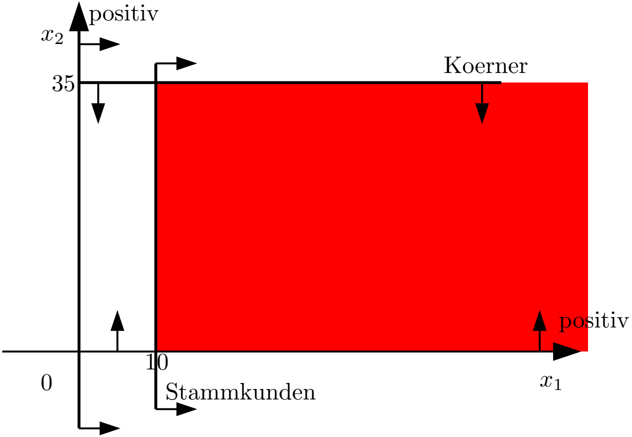
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



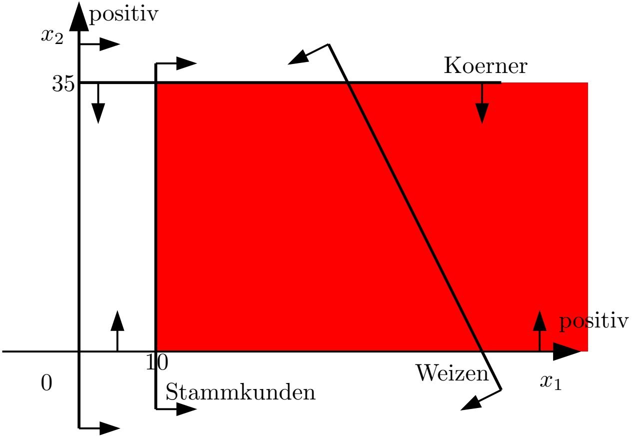
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



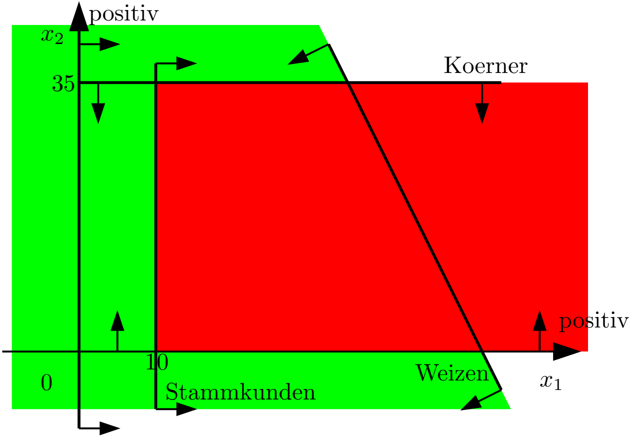
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



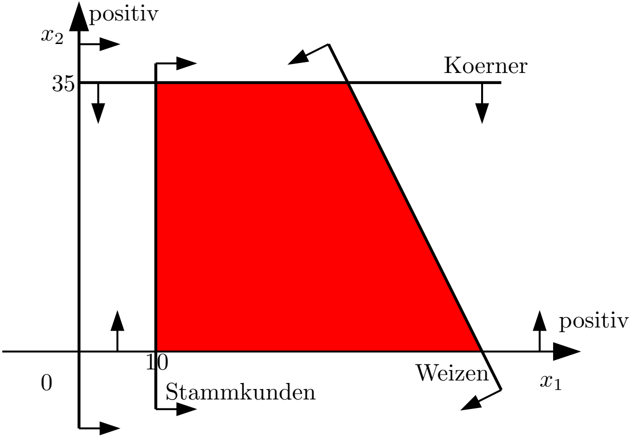
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



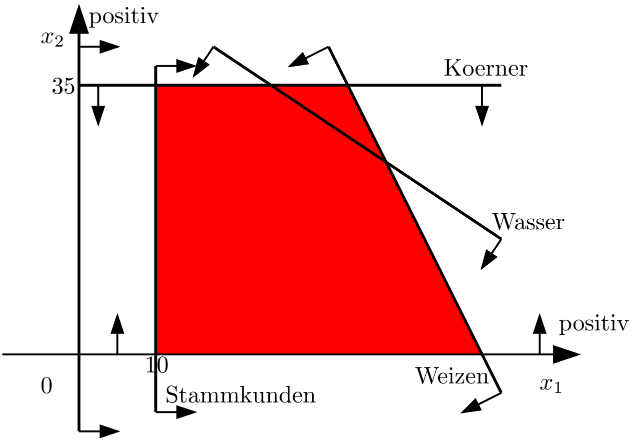
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



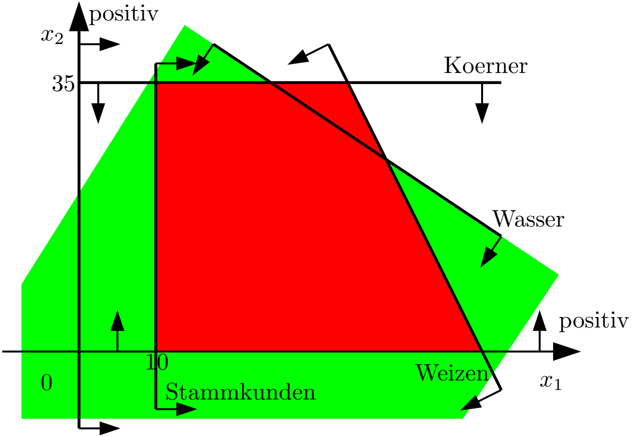
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



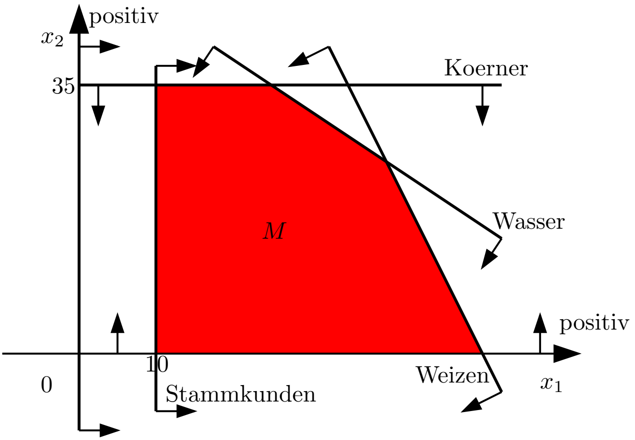
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



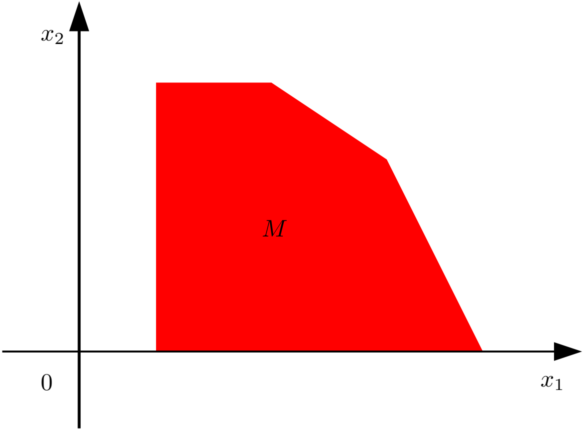
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



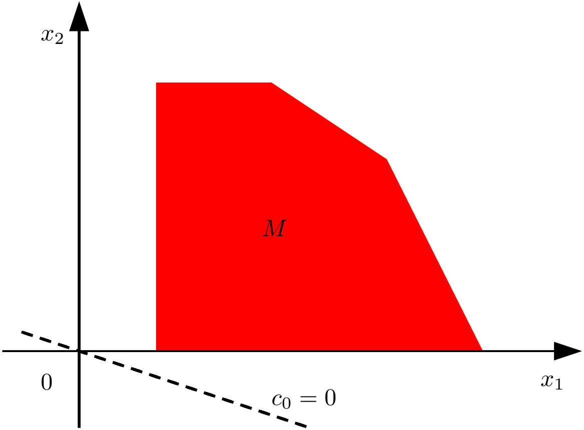
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



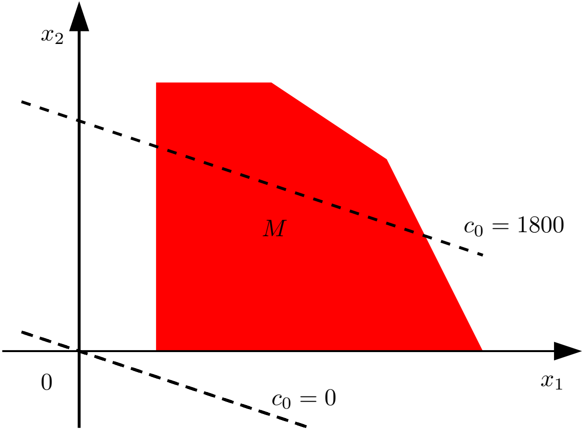
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



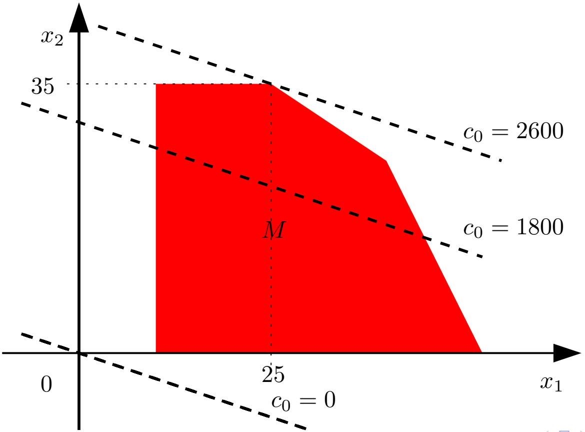
Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



Beispiel: Einfache Bäckersrechnung
Graphischer Lösungsansatz



Beispiel: Einfache Bäckersrechnung
Systematisierung

- *Freie* Variablen $\Longrightarrow$ Differenz zweier nichtnegativer Variablen:

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

Beispiel: Einfache Bäckersrechnung
Systematisierung

- *Freie* Variablen $\Longrightarrow$ Differenz zweier nichtnegativer Variablen:

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

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

$$a_1x_1 + \cdots + a_nx_n \leq b$$

$$\Leftrightarrow$$

$$a_1x_1 + \cdots + a_nx_n + x_{n+1} = b, x_{n+1} \geq 0$$

Beispiel: Einfache Bäckersrechnung
Systematische Mathematische Formulierung

Normalform der linearen Optimierungsaufgabe

$$
\begin{array}{llcl}
\textbf{ZF:} & f(\vec{x}) = 20x_1 + 60x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 & = & \max! \\
\textbf{NB:} & 12x_1 + 6x_2 + x_3 & = & 630 \\
 & 8x_1 + 12x_2 + x_4 & = & 620 \\
 & 10x_2 + x_5 & = & 350 \\
 & x_1 + x_6 & = & 10
\end{array}
$$

$x_i \geq 0$

Normalform
...oder kanonische Form

$$
\begin{array}{rlcl}
\textbf{ZF:} & f(\vec{x}) = c_1 x_1 \dots c_{n-m} x_{n-m} + c_0 & & = \max! \\
\textbf{NB:} & a_{1,1} x_1 + \dots + a_{1,n-m} x_{n-m} + x_{n-m+1} & & = b_1 \\
& \vdots \qquad \vdots \qquad \vdots & \ddots & \vdots \\
& a_{m,1} x_m + \dots + a_{m,n-m} x_{n-m} & + x_n & = b_m \\
& x_i \geq 0 & &
\end{array}
$$

Begriffe

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen

Begriffe

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- *M* ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume

Begriffe

- *Zulässigkeitsbereich M*: alle Punkte, die **NB** erfüllen
- M ist *polyedrisch* i.e. endlicher Schnitt abg. Halbräume
- Ist M zusätzlich beschränkt: *Polytop*

Begriffe

- *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

Begriffe

- *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

Begriffe

- *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
  $(\forall \vec{x_1} \neq \vec{x_2} \in M : \vec{x} \neq \lambda \vec{x_1} + (1-\lambda)\vec{x_2}, 0 < \lambda < 1)$

Begriffe

- *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

Lösbarkeit von Linearen Programmen

(LP) $$\boxed{\begin{array}{l} f(x) =< x, p >= \text{max!} \\ Ax = b \\ x \geq 0 \end{array}}$$

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

Lösbarkeit von Linearen Programmen

(LP) $$\boxed{\begin{array}{l} f(x) = < x, p > = \max! \\ Ax = b \\ x \geq 0 \end{array}}$$

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

Unlösbar falls:

- $M = \emptyset$

Lösbarkeit von Linearen Programmen

(LP) $$\boxed{\begin{array}{l} f(x) = < x, p > = \text{max!} \\ Ax = b \\ x \geq 0 \end{array}}$$

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

Unlösbar falls:

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

Lösbarkeit von Linearen Programmen

(LP) $\boxed{\begin{array}{l} f(x) = \langle x, p \rangle = \text{max!} \\ Ax = b \\ x \geq 0 \end{array}}$

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

Unlösbar falls:

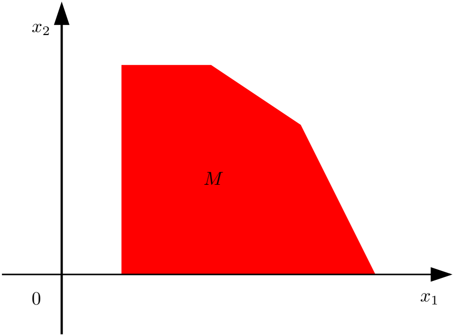
- $M = \emptyset$
- $\sup_{x \in M} f(x) = \infty$

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

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

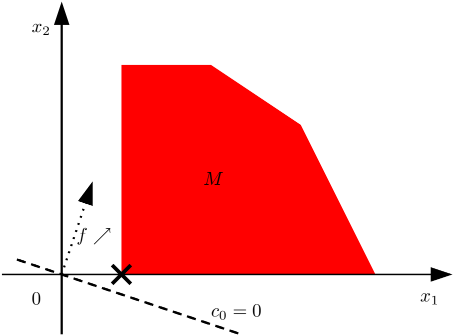
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung.



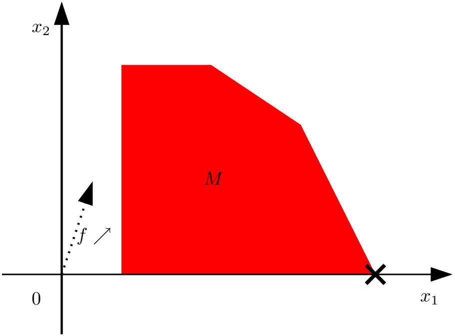
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung.
- Beliebige Startecke x.



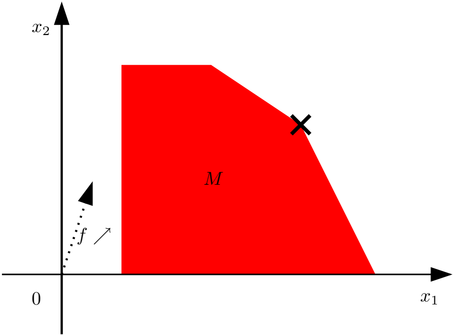
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung.
- Beliebige Startecke x.
- Suche Nachbarecke $\overline{x}$ mit $f(\overline{x}) > f(x)$.



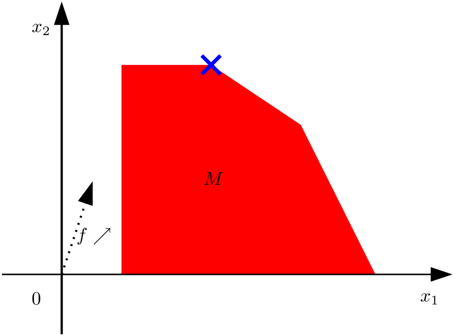
Anschauliche Idee

- Bekannt: Mindestens eine Ecke ist Lösung.
- Beliebige Startecke x.
- Suche Nachbarecke $\overline{x}$ mit $f(\overline{x}) > f(x)$.



Anschauliche Idee

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



Praktische Berechnung

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$

Praktische Berechnung

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:

Praktische Berechnung

- Erstelle erstes *Simplextableau* mit Startecke $(\vec{0}, \vec{b})$
- Man liest am Tableau ab:
  1. Keine weitere Verbesserung ist möglich $\Rightarrow$ Optimum

Praktische Berechnung

- 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

Praktische Berechnung

- 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

Praktische Berechnung

- 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

Praktische Berechnung

- 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.

Probleme bei praktischen Berechnung

- Keine Startecke bekannt. Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, $\Rightarrow$ Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...

Probleme bei praktischen Berechnung

- Keine Startecke bekannt. Einführung von (u.U. vielen) Hilfsvariablen, Herausarbeiten der Hilfsvariablen, ⇒ Startecke; Nordwestecken-Regel; Vogelsche Approximationsmethode...
- Entartete Ecken.

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 $\vec{b}^1 = \vec{b} + \vec{\varepsilon}$

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.

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)

Anwendung des Algorithmus

- Flussprobleme

Anwendung des Algorithmus

- Flussprobleme
- Zuordnungsprobleme

Anwendung des Algorithmus

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme

Anwendung des Algorithmus

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme

Anwendung des Algorithmus

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- ...

Anwendung des Algorithmus

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs! (1/3 ?)

Anwendung des Algorithmus

- Flussprobleme
- Zuordnungsprobleme
- Transportprobleme
- Rundreiseprobleme
- Optimale Strategien bei Spielen
- ...
- Signifikanter Teil der weltweiten Rechenleistung für LPs! (1/3 ?)
- Software: *CPLEX*, *XPRESS*

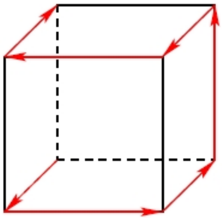
Laufzeiten des Simplex

Laufzeiten des Simplex

- Zunächst: exponentiell

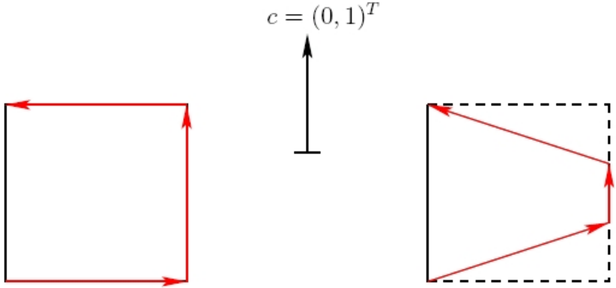
Laufzeiten des Simplex

- Zunächst: exponentiell



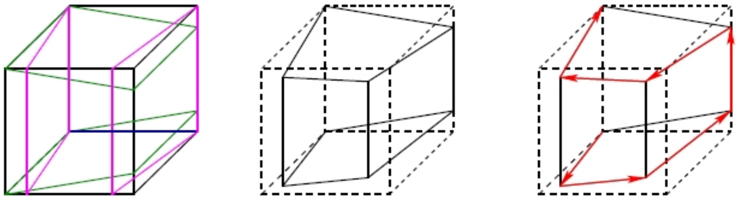
Laufzeiten des Simplex

- Zunächst: exponentiell



Laufzeiten des Simplex

- Zunächst: exponentiell



Laufzeiten des Simplex

- Zunächst: exponentiell 2ex
- $\exists$ ähnliche Beispiele für die meisten Varianten

Laufzeiten des Simplex

- Zunächst: exponentiell 2ex
- $\exists$ ähnliche Beispiele für die meisten Varianten
- $\exists$ Varianten: exponentielle Instanz unbekannt

Laufzeiten des Simplex

- Zunächst: exponentiell 2ex
- $\exists$ ähnliche Beispiele für die meisten Varianten
- $\exists$ Varianten: exponentielle Instanz unbekannt
- $\Rightarrow$ Nicht bekannt ob Simplex polynomial ist

Laufzeiten des Simplex

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

Alternativen zum Simplex

Theorem (Khachian 1979)

LP können in polynomialer Zeit gelöst werden.

Alternativen zum Simplex

Theorem (Khachian 1979)

LP können in polynomialer Zeit gelöst werden.

- K. zeigte, dass der von Shor 1977 vorgestellte *Ellipsoid-Algorithmus* polynomiale Laufzeit hat.

Alternativen zum 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.

Alternativen zum 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*.

Alternativen zum 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 von Farkas

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* $y \in \mathbb{R}^m$

Lemma von Farkas

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

Lemma von Farkas

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

Lemma von Farkas

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

Lemma von Farkas

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

Lemma von Farkas

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

Dualität
Primalprogramm PP vs. Dualprogramm DP

(PP)
$$\boxed{\begin{aligned} &f(x) = x^T p = \max! \\ &Ax = b \\ &x \geq 0 \end{aligned}}$$

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

(DP)
$$\boxed{\begin{aligned} &g(u) = u^T b = \min! \\ &A^T u = p \\ &u \geq 0 \end{aligned}}$$

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

Dualität

Dualprogramme für andere Formen von LP

(PP)

$$f(\vec{x}) = \vec{x}_1^T \vec{p}_1 + \vec{x}_2^T \vec{p}_2 = \text{max!}$$
$$A_{1,1}\vec{x}_1 + A_{1,2}\vec{x}_2 \leq b_1$$
$$A_{2,1}\vec{x}_1 + A_{2,2}\vec{x}_2 = b_2$$
$$\vec{x}_1 \geq 0, \qquad \vec{x}_2 \text{ frei}$$

(DP)

$$f(\vec{u}) = \vec{u}_1^T \vec{b}_1 + \vec{u}_2^T \vec{b}_2 = \text{min!}$$
$$A_{1,1}^T\vec{u}_1 + A_{1,2}^T\vec{u}_2 \leq p_1$$
$$A_{2,1}^T\vec{u}_1 + A_{2,2}^T\vec{u}_2 = p_2$$
$$\vec{u}_1 \geq 0, \qquad \vec{u}_2 \text{ frei}$$

Dualitätssätze

Theorem

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

1. *Schwacher Dualitätssatz:*

Dualitätssätze

Theorem

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

1. *Schwacher Dualitätssatz:*
   1.1 $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.

Dualitätssätze

Theorem

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

1. *Schwacher Dualitätssatz:*
   1.1 $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
   1.2 $\exists x \in M, u \in N \Rightarrow$ **(PP)** *und* **(DP)** *lösbar.*

Dualitätssätze

Theorem

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

1. *Schwacher Dualitätssatz:*
   1.1 $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
   1.2 $\exists x \in M, u \in N \Rightarrow$ **(PP)** *und* **(DP)** *lösbar.*
   1.3 $x_0 \in M, u_0 \in N, f(x) = g(u) \Rightarrow x_0, u_0$ *sind Lösungen.*

Dualitätssätze

Theorem

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

1. *Schwacher Dualitätssatz:*
   1.1 $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
   1.2 $\exists x \in M, u \in N \Rightarrow$ **(PP)** *und* **(DP)** *lösbar.*
   1.3 $x_0 \in M, u_0 \in N, f(x) = g(u) \Rightarrow x_0, u_0$ *sind Lösungen.*
2. *Starker Dualitätssatz:*

Dualitätssätze

Theorem

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

1. *Schwacher Dualitätssatz:*
   1.1 $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
   1.2 $\exists x \in M, u \in N \Rightarrow$ **(PP)** *und* **(DP)** *lösbar.*
   1.3 $x_0 \in M, u_0 \in N, f(x) = g(u) \Rightarrow x_0, u_0$ *sind Lösungen.*
2. *Starker Dualitätssatz:*
   2.1 **(PP)** *lösbar* $\Rightarrow$ **(DP)** *lösbar*
   und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Dualitätssätze

Theorem

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

1. *Schwacher Dualitätssatz:*
   - 1.1 $x \in M, u \in N \Rightarrow f(x) \leq g(u)$.
   - 1.2 $\exists x \in M, u \in N \Rightarrow$ **(PP)** *und* **(DP)** *lösbar.*
   - 1.3 $x_0 \in M, u_0 \in N, f(x) = g(u) \Rightarrow x_0, u_0$ *sind Lösungen.*
2. *Starker Dualitätssatz:*
   - 2.1 **(PP)** *lösbar* $\Rightarrow$ **(DP)** *lösbar*
     und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$
   - 2.2 **(PP)** *lösbar* $\Rightarrow$ **(DP)** *lösbar*
     und: $\max_{x \in M} f(x) = \min_{u \in N} g(u)$

Anwendungen der Dualitätssätze

- Normalform für ein duales Problem leichter zu finden

Anwendungen der Dualitätssätze

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

Allgemeines

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)

Allgemeines

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)
- *Relaxierung* ILP $\rightarrow$ LP (ohne Ganzzahligkeitsbedingung)

Allgemeines

- Ganzzahlige Lineare Programme (*ILP*) sind NP-schwer (Bsp.: TSP)
- *Relaxierung* ILP $\rightarrow$ LP (ohne Ganzzahligkeitsbedingung)
  Dann: Durch Runden (suboptimale) ganzzahlige Lösung finden

Allgemeines

- 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 ($\notin M$)

Allgemeines

- 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?

Total unimodulare Matrizen

Definition

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

Total unimodulare Matrizen

Definition

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

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

Total unimodulare Matrizen

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.*

Total unimodulare Matrizen

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!

Beweis des Ganzzahligkeitssatzes

1. $Ax \leq b, x \geq 0$, b ganzz., A TUM

Beweis des Ganzzahligkeitssatzes

1. $Ax \leq b, x \geq 0$, b ganzz., A TUM
2. Schlupfvariablen: $b = [A \quad I]x^I, x^I \in \mathbb{R}^{n+m}$

Beweis des Ganzzahligkeitssatzes

1. $Ax \leq b, x \geq 0$, b ganzz., A TUM
2. Schlupfvariablen: $b = [A \quad I]x^I, x^I \in \mathbb{R}^{n+m}$
3. Lemma: A TUM $\Rightarrow [A \quad I]$

Beweis des Ganzzahligkeitssatzes

1. $Ax \leq b, x \geq 0$, b ganzz., A TUM
2. Schlupfvariablen: $b = [A \quad I]x^I, x^I \in \mathbb{R}^{n+m}$
3. Lemma: A TUM $\Rightarrow [A \quad I]$
4. SA: Jede Ecke $\overline{x}$ festgelegt durch $\overline{A}\overline{x} = \overline{b}$
   $\overline{A} \in \mathbb{R}^{m \times m}$ Submatrix von $[A \quad I]$, $\overline{x}$ hat $n \times 0$
   mit $\overline{A}$ TUM, $\det \overline{A} \in \{1, -1\}$ (nichtsingulär)

Beweis des Ganzzahligkeitssatzes

1. $Ax \leq b, x \geq 0$, b ganzz., A TUM
2. Schlupfvariablen: $b = [A \quad I]x^I, x^I \in \mathbb{R}^{n+m}$
3. Lemma: A TUM $\Rightarrow [A \quad I]$
4. SA: Jede Ecke $\overline{x}$ festgelegt durch $\overline{A}\overline{x} = \overline{b}$
   $\overline{A} \in \mathbb{R}^{m \times m}$ Submatrix von $[A \quad I]$, $\overline{x}$ hat $n \times 0$
   mit $\overline{A}$ TUM, $\det \overline{A} \in \{1, -1\}$ (nichtsingulär)
5. Cramersche Regel:
   $$\overline{x}_j = \frac{\det B^{(j)}}{\det \overline{A}}$$
   (wobei $B^{(j)}$ $\overline{A}$ ist, mit $\overline{b}$ als j. Spalte)

Beweis des Ganzzahligkeitssatzes

1. $Ax \leq b, x \geq 0$, b ganzz., A TUM
2. Schlupfvariablen: $b = [A \quad I]x^I, x^I \in \mathbb{R}^{n+m}$
3. Lemma: A TUM $\Rightarrow [A \quad I]$
4. SA: Jede Ecke $\overline{x}$ festgelegt durch $\overline{A}\overline{x} = \overline{b}$
   $\overline{A} \in \mathbb{R}^{m \times m}$ Submatrix von $[A \quad I]$, $\overline{x}$ hat $n \times 0$
   mit $\overline{A}$ TUM, $\det \overline{A} \in \{1, -1\}$ (nichtsingulär)
5. Cramersche Regel:
   $$\overline{x}_j = \frac{\det B^{(j)}}{\det \overline{A}}$$
   (wobei $B^{(j)}$ $\overline{A}$ ist, mit $\overline{b}$ als j. Spalte)
6. $\Rightarrow \overline{x}_j$ ist ganzzahlig.

Anwendungen von totaler Unimodularität
Matching

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

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

Anwendungen von totaler Unimodularität
Matching

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

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

Lösung durch ein ILP ?

Anwendungen von totaler Unimodularität
Ein schöner Satz

Theorem
Inizidenzmatrix A TUM $\Longleftrightarrow$ *Graph bipartit*

Anwendungen von totaler Unimodularität
Ein schöner Satz

Theorem

Inizidenzmatrix A TUM $\Longleftrightarrow$ *Graph bipartit*

Beweis.

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

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

$\Rightarrow$ Widerspruch

$\square$

Anwendungen von totaler Unimodularität
Ein schöner Satz

Theorem

Inizidenzmatrix A TUM $\Longleftrightarrow$ *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 $\square$

Anwendungen von totaler Unimodularität
ILP Probleme die eigentlich LP sind

- Somit ist das matching Problem als LP lösbar

Anwendungen von totaler Unimodularität
ILP Probleme die eigentlich LP sind

- Somit ist das matching Problem als LP lösbar
- Auch Inzidenzmatritzen von gerichteten Graphen snd TUM

Anwendungen von totaler Unimodularität
ILP Probleme die eigentlich LP sind

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

Ende

Danke für Eure Aufmerksamkeit!

Ende

Danke für Eure Aufmerksamkeit!

Fragen?