Effiziente Algorithmen (SS2022)
Chapter 4
Lineare Programme 1

Walter Unger

Lehrstuhl für Informatik 1

17.05.2022 17:57:45

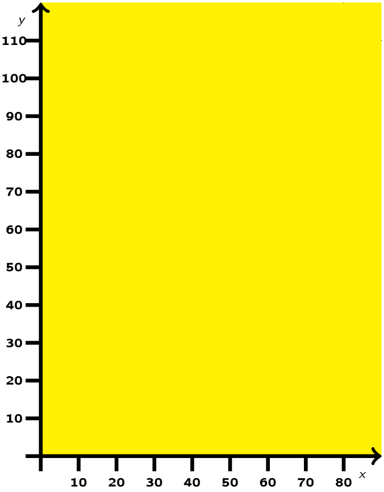
Contents I

Einfaches Beispiel

Einfaches Beispiel

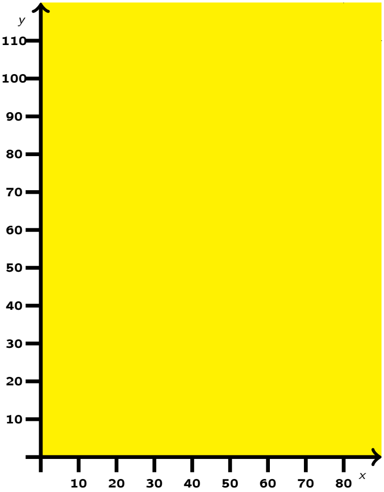
- Wir betrachten ein System von linearen Ungleichungen.

Einfaches Beispiel



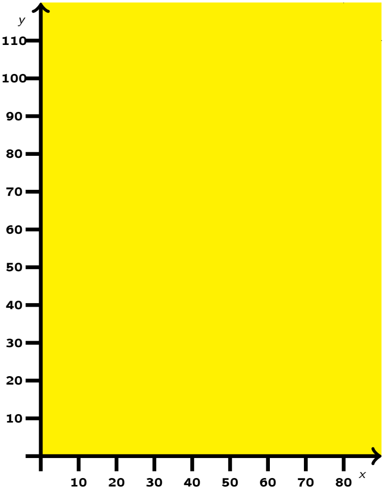
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.

Einfaches Beispiel



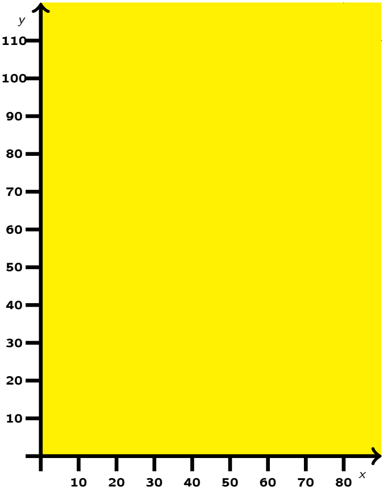
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine “Zielfunktion” zu optimieren.
- Beispiel (Brotrezept):

Einfaches Beispiel



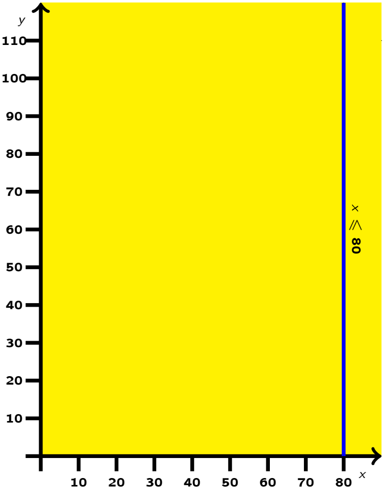
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl

Einfaches Beispiel



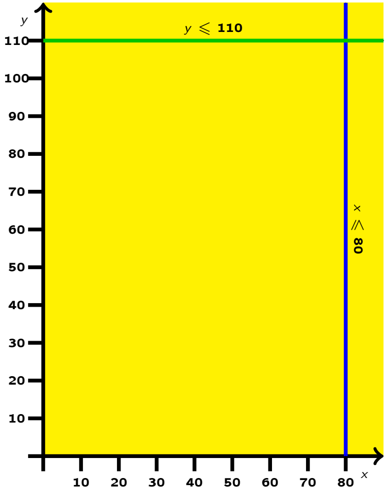
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl

Einfaches Beispiel



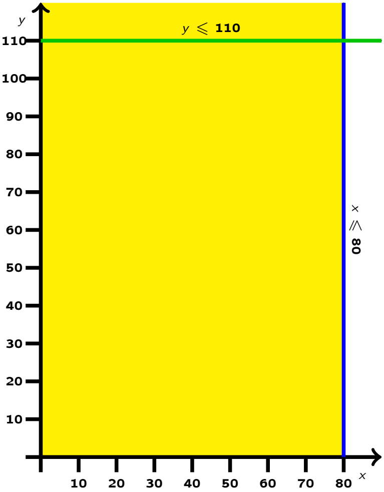
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.

Einfaches Beispiel



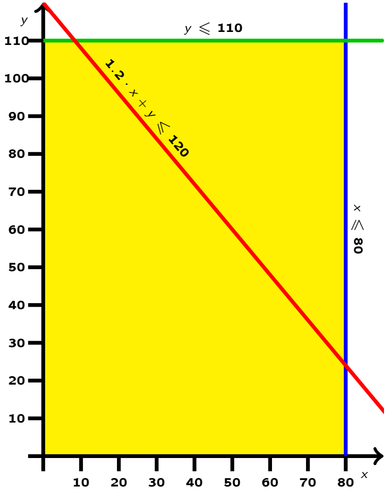
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.

Einfaches Beispiel



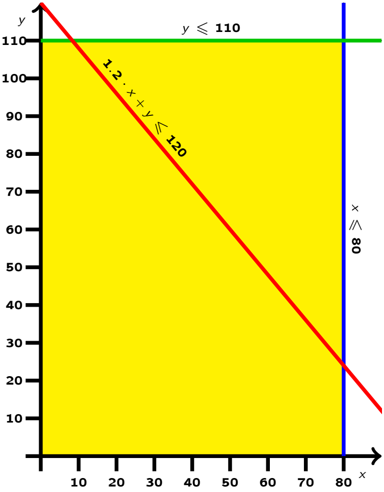
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine “Zielfunktion” zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.

Einfaches Beispiel



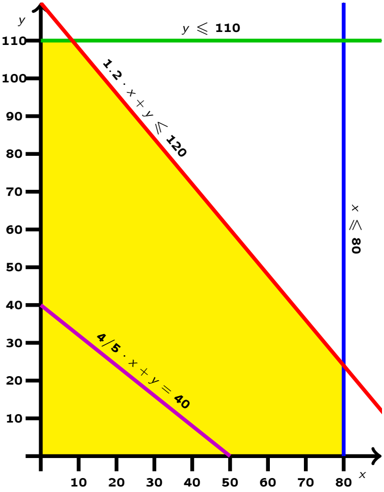
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine “Zielfunktion” zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.

Einfaches Beispiel



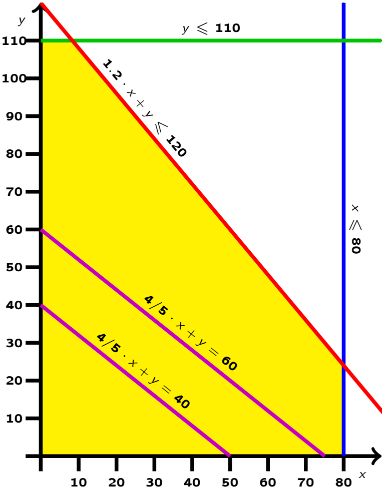
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Einfaches Beispiel



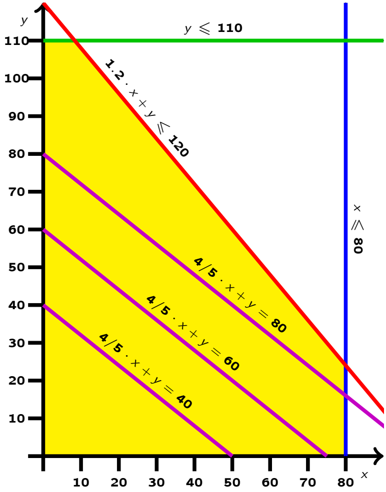
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Einfaches Beispiel



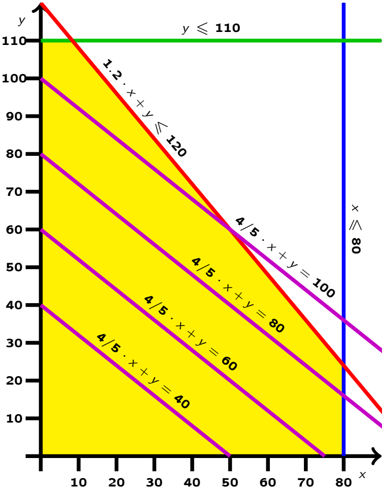
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Einfaches Beispiel



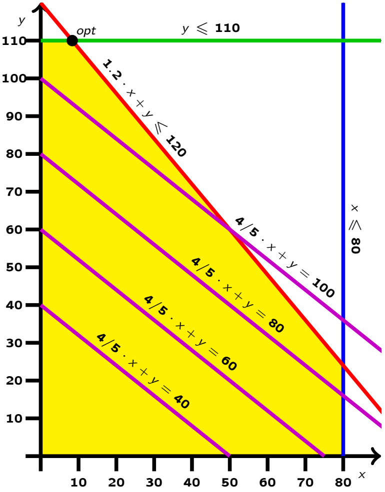
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Einfaches Beispiel



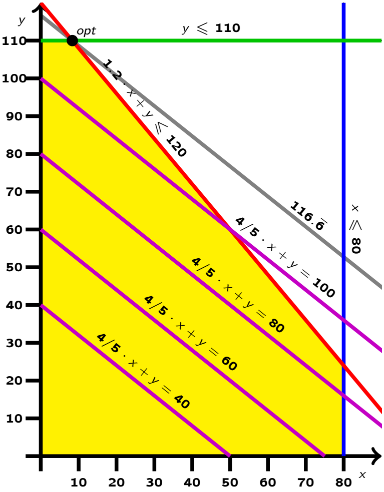
- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Einfaches Beispiel



- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Einfaches Beispiel



- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Beispiel: Flussproblem

- Flussproblem:

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.
  - Maximiere

$$\sum_{e \in N_{out}(s) \in E} x_e.$$

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.
  - Maximiere
$$\sum_{e \in N_{out}(s) \in E} x_e.$$
  - unter Einhaltung der Bedingungen:

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.
  - Maximiere
    $$\sum_{e \in N_{out}(s) \in E} x_e.$$
  - unter Einhaltung der Bedingungen:
    - Für jeden Knoten $v \in V \setminus \{s, t\}$: $\sum_{e \in N_{in}(v)} x_e = \sum_{e \in N_{out}(v)} x_e$,

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.
  - Maximiere
    $$\sum_{e \in N_{out}(s) \in E} x_e.$$
  - unter Einhaltung der Bedingungen:
    - Für jeden Knoten $v \in V \setminus \{s, t\}$: $\sum_{e \in N_{in}(v)} x_e = \sum_{e \in N_{out}(v)} x_e$,
    - $\forall e \in E : x_e \leqslant c_e$, und

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.
  - Maximiere
$$\sum_{e \in N_{out}(s) \in E} x_e.$$
  - unter Einhaltung der Bedingungen:
    - Für jeden Knoten $v \in V \setminus \{s, t\}$: $\sum_{e \in N_{in}(v)} x_e = \sum_{e \in N_{out}(v)} x_e$,
    - $\forall e \in E : x_e \leqslant c_e$, und
    - $\forall e \in E : x_e \geqslant 0$.

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.
  - Maximiere
    $$\sum_{e \in N_{out}(s) \in E} x_e.$$
  - unter Einhaltung der Bedingungen:
    - Für jeden Knoten $v \in V \setminus \{s, t\}$: $\sum_{e \in N_{in}(v)} x_e = \sum_{e \in N_{out}(v)} x_e$,
    - $\forall e \in E : x_e \leqslant c_e$, und
    - $\forall e \in E : x_e \geqslant 0$.

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:
  - Maximiere

$$\sum_{i=1}^{d} v_i \cdot x_i.$$

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:
  - Maximiere

$$\sum_{i=1}^{d} v_i \cdot x_i.$$

  - unter den Nebenbedingungen:

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:
  - Maximiere
$$\sum_{i=1}^{d} v_i \cdot x_i.$$
  - unter den Nebenbedingungen:
    - $\sum_{i=1}^{d} g_i \cdot x_i \leqslant G$,

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:
  - Maximiere
$$\sum_{i=1}^{d} v_i \cdot x_i.$$
  - unter den Nebenbedingungen:
    - $\sum_{i=1}^{d} g_i \cdot x_i \leqslant G$,
    - $\forall i : 1 \leqslant i \leqslant d : x_i \leqslant 1$, und

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:
  - Maximiere
$$\sum_{i=1}^{d} v_i \cdot x_i.$$
  - unter den Nebenbedingungen:
    - $\sum_{i=1}^{d} g_i \cdot x_i \leqslant G$,
    - $\forall i : 1 \leqslant i \leqslant d : x_i \leqslant 1$, und
    - $\forall i : 1 \leqslant i \leqslant d : x_i \geqslant 0$.

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:
  - Maximiere
$$\sum_{i=1}^{d} v_i \cdot x_i.$$
  - unter den Nebenbedingungen:
    - $\sum_{i=1}^{d} g_i \cdot x_i \leqslant G$,
    - $\forall i : 1 \leqslant i \leqslant d : x_i \leqslant 1$, und
    - $\forall i : 1 \leqslant i \leqslant d : x_i \geqslant 0$.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
    - N_E Endknoten.
    - N_R Router mit Konverter.
    - $n = N_E + N_R$.
    - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
    - m Anzahl der Lichtwege.
    - E: $(i, j) \in E \iff$ Kante von N_i nach N_j.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.
  - E: $(i, j) \in E \iff$ Kante von N_i nach N_j.
  - $src(k)$: ist Startknoten der k-ten Anfrage.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.
  - E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
  - $src(k)$: ist Startknoten der k-ten Anfrage.
  - $dst(k)$: ist Endknoten der k-ten Anfrage.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.
  - E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
  - $src(k)$: ist Startknoten der k-ten Anfrage.
  - $dst(k)$: ist Endknoten der k-ten Anfrage.
  - Ω_{max}: Congestion des Netzwerks.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.
  - E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
  - $src(k)$: ist Startknoten der k-ten Anfrage.
  - $dst(k)$: ist Endknoten der k-ten Anfrage.
  - Ω_{max}: Congestion des Netzwerks.
  - $X_{ij}^k \in \{0,1\}$ mit:

$$X_{ij}^k = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt Kante } (i,j) \in E \\ 0 & \text{sonst} \end{cases}$$

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.
  - E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
  - $src(k)$: ist Startknoten der k-ten Anfrage.
  - $dst(k)$: ist Endknoten der k-ten Anfrage.
  - Ω_{max}: Congestion des Netzwerks.
  - $X_{ij}^k \in \{0,1\}$ mit:

$$X_{ij}^k = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt Kante } (i,j) \in E \\ 0 & \text{sonst} \end{cases}$$

  - Wegen $X_{ij}^k \in \{0,1\}$ ist dies hier ein “Integer Linerar Programm”.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.
  - E: $(i, j) \in E \iff$ Kante von N_i nach N_j.
  - $src(k)$: ist Startknoten der k-ten Anfrage.
  - $dst(k)$: ist Endknoten der k-ten Anfrage.
  - Ω_{max}: Congestion des Netzwerks.
  - $X_{ij}^k \in \{0, 1\}$ mit:

$$X_{ij}^k = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt Kante } (i,j) \in E \\ 0 & \text{sonst} \end{cases}$$

  - Wegen $X_{ij}^k \in \{0, 1\}$ ist dies hier ein "Integer Linerar Programm".

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (1 \leqslant k \leqslant m)$
X_{ij}^k : k**-te Weg nutzt Kante** (i,j)

- Minimiere Zielfunktion Ω_{max}.

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (1 \leqslant k \leqslant m)$
X_{ij}^k : k-te **Weg nutzt Kante** (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^m X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (\mathbf{1} \leqslant k \leqslant m)$
X_{ij}^k : k**-te Weg nutzt Kante** (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^m X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (1 \leqslant k \leqslant m)$
X_{ij}^k : k-**te Weg nutzt Kante** (i, j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^{m} X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (1 \leqslant k \leqslant m)$
X_{ij}^k : k-te Weg nutzt Kante (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^m X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.
- Komplexität:

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (\mathbf{1} \leqslant k \leqslant m)$
X_{ij}^k : k**-te Weg nutzt Kante** (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^m X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.
- Komplexität:
  - $m \cdot |E|$ Variablen der Form X_{ij}^k.

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k)$: $(1 \leqslant k \leqslant m)$
X_{ij}^k : k-te Weg nutzt Kante (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^{m} X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.
- Komplexität:
  - $m \cdot |E|$ Variablen der Form X_{ij}^k.
  - Eine Variable Ω_{max}.

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (1 \leqslant k \leqslant m)$
X_{ij}^k : k-te Weg nutzt Kante (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^{m} X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.
- Komplexität:
  - $m \cdot |E|$ Variablen der Form X_{ij}^k.
  - Eine Variable Ω_{max}.
  - Nebenbedingungen: $|E| + n \cdot m$.

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k)$: $(1 \leqslant k \leqslant m)$
X_{ij}^k : k-te Weg nutzt Kante (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^{m} X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.
- Komplexität:
  - $m \cdot |E|$ Variablen der Form X_{ij}^k.
  - Eine Variable Ω_{max}.
  - Nebenbedingungen: $|E| + n \cdot m$.
  - Schon für relativ kleine Netzwerke zu aufwendig.

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k)$: $(1 \leqslant k \leqslant m)$
X_{ij}^k : **k-te Weg nutzt Kante** (i,j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^m X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.
- Komplexität:
  - $m \cdot |E|$ Variablen der Form X_{ij}^k.
  - Eine Variable Ω_{max}.
  - Nebenbedingungen: $|E| + n \cdot m$.
  - Schon für relativ kleine Netzwerke zu aufwendig.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i,j) \in E \iff$ Kante von N_i nach N_j.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i,j) \in E \Longleftrightarrow$ Kante von N_i nach N_j.
- $src(k)$: ist Startknoten der k-ten Anfrage.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
- $src(k)$: ist Startknoten der k-ten Anfrage.
- $dst(k)$: ist Endknoten der k-ten Anfrage.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i, j) \in E \iff$ Kante von N_i nach N_j.
- $src(k)$: ist Startknoten der k-ten Anfrage.
- $dst(k)$: ist Endknoten der k-ten Anfrage.
- Ω_{max}: Congestion des Netzwerks.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
- $src(k)$: ist Startknoten der k-ten Anfrage.
- $dst(k)$: ist Endknoten der k-ten Anfrage.
- Ω_{max}: Congestion des Netzwerks.
- $X_{ij}^{wk} \in \{0,1\}$ mit:

$$X_{ij}^{wk} = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt } (i,j) \in E \text{ und W.länge } w \\ 0 & \text{sonst} \end{cases}$$

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
- $src(k)$: ist Startknoten der k-ten Anfrage.
- $dst(k)$: ist Endknoten der k-ten Anfrage.
- Ω_{max}: Congestion des Netzwerks.
- $X_{ij}^{wk} \in \{0,1\}$ mit:

$$X_{ij}^{wk} = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt } (i,j) \in E \text{ und W.länge } w \\ 0 & \text{sonst} \end{cases}$$

- $Y_w^k \in \{0,1\}$ mit:

$$Y_w^k = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt Wellenlänge } w \\ 0 & \text{sonst} \end{cases}$$

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
- $src(k)$: ist Startknoten der k-ten Anfrage.
- $dst(k)$: ist Endknoten der k-ten Anfrage.
- Ω_{max}: Congestion des Netzwerks.
- $X_{ij}^{wk} \in \{0,1\}$ mit:

$$X_{ij}^{wk} = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt } (i,j) \in E \text{ und W.länge } w \\ 0 & \text{sonst} \end{cases}$$

- $Y_w^k \in \{0,1\}$ mit:

$$Y_w^k = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt Wellenlänge } w \\ 0 & \text{sonst} \end{cases}$$

ILP:

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (1 \leqslant k \leqslant m)$
X_{ij}^{k} : k-te **Weg nutzt Kante** (i,j)
Y_{w}^{k} : k-te **Weg nutzt Wellenlänge** w

- Minimiere Zielfunktion Ω_{max}:

$$\sum_{w=1}^{n_{ch}} \sum_{k=1}^{m} X_{ij}^{wk} \leqslant \Omega_{max}, \forall (i,j) \in E$$

ILP:

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k)$: $(1 \leqslant k \leqslant m)$
X_{ij}^k : k**-te Weg nutzt Kante** (i,j)
Y_w^k : k**-te Weg nutzt Wellenlänge** w

- Minimiere Zielfunktion Ω_{max}:

$$\sum_{w=1}^{n_{ch}} \sum_{k=1}^{m} X_{ij}^{wk} \leqslant \Omega_{max}, \forall (i,j) \in E$$

- Für alle $k, w : 1 \leqslant k \leqslant m, 1 \leqslant w \leqslant n_{ch}$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^{wk} - \sum_{j:(j,i)\in E} X_{ji}^{wk} = \begin{cases} Y_w^k & \text{falls } src(k) = i \\ -Y_w^k & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

ILP:

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (\mathbf{1} \leqslant k \leqslant m)$
X_{ij}^k : k-**te Weg nutzt Kante** (i,j)
Y_w^k : k-**te Weg nutzt Wellenlänge** w

- Minimiere Zielfunktion Ω_{max}:

$$\sum_{w=1}^{n_{ch}} \sum_{k=1}^{m} X_{ij}^{wk} \leqslant \Omega_{max}, \forall (i,j) \in E$$

- Für alle $k, w : 1 \leqslant k \leqslant m, 1 \leqslant w \leqslant n_{ch}$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^{wk} - \sum_{j:(j,i)\in E} X_{ji}^{wk} = \begin{cases} Y_w^k & \text{falls } src(k) = i \\ -Y_w^k & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Für alle $k : 1 \leqslant k \leqslant m$:

$$\sum_{w=1}^{n_{ch}} Y_w^k = 1$$

ILP:

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k)$: $(1 \leqslant k \leqslant m)$
X_{ij}^k : k-te Weg nutzt Kante (i,j)
Y_w^k : k-te Weg nutzt Wellenlänge w

- Minimiere Zielfunktion Ω_{max}:

$$\sum_{w=1}^{n_{ch}} \sum_{k=1}^{m} X_{ij}^{wk} \leqslant \Omega_{max}, \forall (i,j) \in E$$

- Für alle $k, w : 1 \leqslant k \leqslant m, 1 \leqslant w \leqslant n_{ch}$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^{wk} - \sum_{j:(j,i)\in E} X_{ji}^{wk} = \begin{cases} Y_w^k & \text{falls } src(k) = i \\ -Y_w^k & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Für alle $k : 1 \leqslant k \leqslant m$:

$$\sum_{w=1}^{n_{ch}} Y_w^k = 1$$

- Für alle $w : 1 \leqslant w \leqslant n_{ch}$ und alle $(i,j) \in E$:

$$\sum_{k=1}^{m} X_{ij}^{wk} \leqslant 1$$

ILP:

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k)$: $(1 \leqslant k \leqslant m)$
X_{ij}^k : k**-te Weg nutzt Kante** (i,j)
Y_w^k : k**-te Weg nutzt Wellenlänge** w

- Minimiere Zielfunktion Ω_{max}:

$$\sum_{w=1}^{n_{ch}} \sum_{k=1}^{m} X_{ij}^{wk} \leqslant \Omega_{max}, \forall (i,j) \in E$$

- Für alle $k, w : 1 \leqslant k \leqslant m, 1 \leqslant w \leqslant n_{ch}$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^{wk} - \sum_{j:(j,i)\in E} X_{ji}^{wk} = \begin{cases} Y_w^k & \text{falls } src(k) = i \\ -Y_w^k & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Für alle $k : 1 \leqslant k \leqslant m$:

$$\sum_{w=1}^{n_{ch}} Y_w^k = 1$$

- Für alle $w : 1 \leqslant w \leqslant n_{ch}$ und alle $(i,j) \in E$:

$$\sum_{k=1}^{m} X_{ij}^{wk} \leqslant 1$$

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.
- Setze:

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.
- Setze:
  - $x = (x_j)$, $c = (c_j)$, $b = (b_i)$ und $A = (a_{ij})$.

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.
- Setze:
  - $x = (x_j)$, $c = (c_j)$, $b = (b_i)$ und $A = (a_{ij})$.
- Dann ist kurzgefasst die kanonische Form:

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.
- Setze:
  - $x = (x_j)$, $c = (c_j)$, $b = (b_i)$ und $A = (a_{ij})$.
- Dann ist kurzgefasst die kanonische Form:
  - Maximiere $c^T \cdot x$ unter den Nebenbedingungen:

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.
- Setze:
  - $x = (x_j)$, $c = (c_j)$, $b = (b_i)$ und $A = (a_{ij})$.
- Dann ist kurzgefasst die kanonische Form:
  - Maximiere $c^T \cdot x$ unter den Nebenbedingungen:
  - $A \cdot x \leqslant b$ und $x \geqslant 0$.

Kanonische Form eines LP

- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_i$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.
- Setze:
  - $x = (x_j)$, $c = (c_j)$, $b = (b_i)$ und $A = (a_{ij})$.
- Dann ist kurzgefasst die kanonische Form:
  - Maximiere $c^T \cdot x$ unter den Nebenbedingungen:
  - $A \cdot x \leqslant b$ und $x \geqslant 0$.

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch
  - $-a^T \cdot x \leqslant -b$.

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch
  - $-a^T \cdot x \leqslant -b$.
- Eine möglicherweise negative Variable $x \in \mathbb{R}$ wird ersetzt durch:

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch
  - $-a^T \cdot x \leqslant -b$.
- Eine möglicherweise negative Variable $x \in \mathbb{R}$ wird ersetzt durch:
  - $x' - x''$ und den Nebenbedingungen:

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch
  - $-a^T \cdot x \leqslant -b$.
- Eine möglicherweise negative Variable $x \in \mathbb{R}$ wird ersetzt durch:
  - $x' - x''$ und den Nebenbedingungen:
  - $x' \geqslant 0$ und

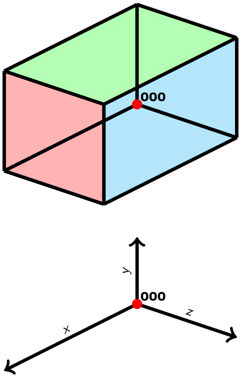
Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch
  - $-a^T \cdot x \leqslant -b$.
- Eine möglicherweise negative Variable $x \in \mathbb{R}$ wird ersetzt durch:
  - $x' - x''$ und den Nebenbedingungen:
  - $x' \geqslant 0$ und
  - $x'' \geqslant 0$.

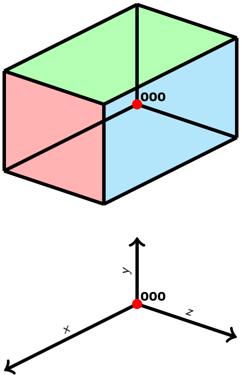
Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch
  - $-a^T \cdot x \leqslant -b$.
- Eine möglicherweise negative Variable $x \in \mathbb{R}$ wird ersetzt durch:
  - $x' - x''$ und den Nebenbedingungen:
  - $x' \geqslant 0$ und
  - $x'' \geqslant 0$.

Geometrische Interpretation



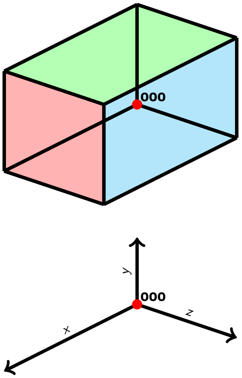
Geometrische Interpretation



$0 \leqslant x \leqslant t$
$0 \leqslant y \leqslant t$
$0 \leqslant z \leqslant t$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.

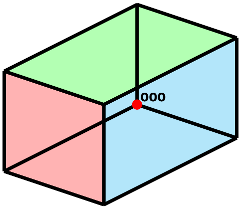
Geometrische Interpretation

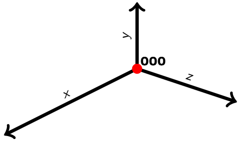


$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \dots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.

Geometrische Interpretation

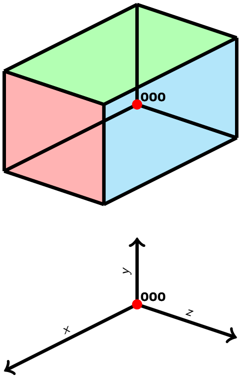




$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.
- Die Grenze des Halbraums ist die Hyperebene $a_i \cdot x = b_i$.

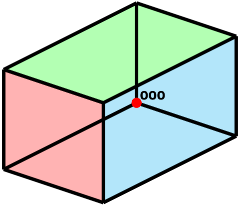
Geometrische Interpretation

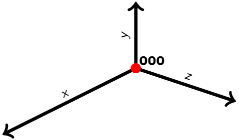


$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.
- Die Grenze des Halbraums ist die Hyperebene $a_i \cdot x = b_i$.
- Der Schnitt aller Halbräume ist der Raum der zulässigen Lösungen.

Geometrische Interpretation

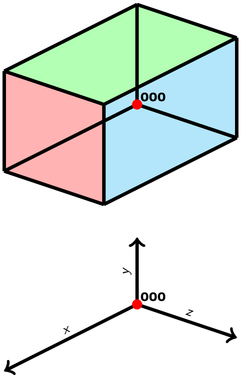




$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.
- Die Grenze des Halbraums ist die Hyperebene $a_i \cdot x = b_i$.
- Der Schnitt aller Halbräume ist der Raum der zulässigen Lösungen.
- Ein LP wird als zulässig bezeichnet, wenn des zulässige Lösungen gibt.

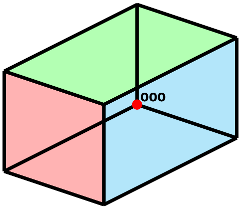
Geometrische Interpretation

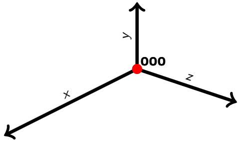


$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.
- Die Grenze des Halbraums ist die Hyperebene $a_i \cdot x = b_i$.
- Der Schnitt aller Halbräume ist der Raum der zulässigen Lösungen.
- Ein LP wird als zulässig bezeichnet, wenn des zulässige Lösungen gibt.
- Schnittmengen von Halbräumen bilden ein Polyhedron.

Geometrische Interpretation

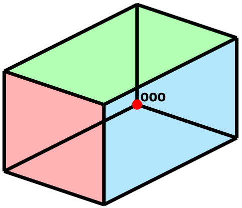


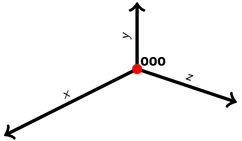


$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.
- Die Grenze des Halbraums ist die Hyperebene $a_i \cdot x = b_i$.
- Der Schnitt aller Halbräume ist der Raum der zulässigen Lösungen.
- Ein LP wird als zulässig bezeichnet, wenn des zulässige Lösungen gibt.
- Schnittmengen von Halbräumen bilden ein Polyhedron.
- Damit bilden die zulässigen Lösungen ein Polyhedron.

Geometrische Interpretation





$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.
- Die Grenze des Halbraums ist die Hyperebene $a_i \cdot x = b_i$.
- Der Schnitt aller Halbräume ist der Raum der zulässigen Lösungen.
- Ein LP wird als zulässig bezeichnet, wenn des zulässige Lösungen gibt.
- Schnittmengen von Halbräumen bilden ein Polyhedron.
- Damit bilden die zulässigen Lösungen ein Polyhedron.

Konvexität

P **Polyhedron**

Lemma

Ein Polyhedron P ist konvex.

Beweis:

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:

Konvexität

P **Polyhedron**

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.

Konvexität

P **Polyhedron**

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.

Konvexität

P **Polyhedron**

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.
- Zeige: Der Schnitt von konvexen Mengen ist konvex.

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.
- Zeige: Der Schnitt von konvexen Mengen ist konvex.
- D.h. aus A, B konvex folgt $A \cap B$ konvex.

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.
- Zeige: Der Schnitt von konvexen Mengen ist konvex.
- D.h. aus A, B konvex folgt $A \cap B$ konvex.
- Damit gilt: $\forall a, b \in A : l(a, b) \in A$ und

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.
- Zeige: Der Schnitt von konvexen Mengen ist konvex.
- D.h. aus A, B konvex folgt $A \cap B$ konvex.
- Damit gilt: $\forall a, b \in A : l(a, b) \in A$ und
- weiter $\forall a, b \in B : l(a, b) \in B$.

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda) b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.
- Zeige: Der Schnitt von konvexen Mengen ist konvex.
- D.h. aus A, B konvex folgt $A \cap B$ konvex.
- Damit gilt: $\forall a, b \in A : l(a, b) \in A$ und
- weiter $\forall a, b \in B : l(a, b) \in B$.
- Es folgt: $\forall a, b \in A \cap B : l(a, b) \in A \cap B$.

Konvexität

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.
- Zeige: Der Schnitt von konvexen Mengen ist konvex.
- D.h. aus A, B konvex folgt $A \cap B$ konvex.
- Damit gilt: $\forall a, b \in A : l(a, b) \in A$ und
- weiter $\forall a, b \in B : l(a, b) \in B$.
- Es folgt: $\forall a, b \in A \cap B : l(a, b) \in A \cap B$.

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

Lokales und globales Optimum

P **Polyhedron**

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

Lokales und globales Optimum

P **Polyhedron**

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.

Lokales und globales Optimum

P **Polyhedron**

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda)z$.

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda) z$.
- Es folgt:

$$c^T y \quad = \quad c^T(\lambda x + (1 - \lambda) z)$$

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda) z$.
- Es folgt:

$$c^T y = c^T(\lambda x + (1 - \lambda) z)$$

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda)z$.
- Es folgt:

$$\begin{aligned} c^T y &= c^T(\lambda x + (1 - \lambda)z) \\ &= \lambda c^T x + (1 - \lambda)c^T z \end{aligned}$$

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda) z$.
- Es folgt:

$$\begin{aligned} c^T y &= c^T(\lambda x + (1-\lambda) z) \\ &= \lambda c^T x + (1-\lambda) c^T z \\ &> \lambda c^T x + (1-\lambda) c^T x \end{aligned}$$

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda) z$.
- Es folgt:

$$\begin{array}{rcl} c^T y & = & c^T(\lambda x + (1-\lambda)z) \\ & = & \lambda c^T x + (1-\lambda) c^T z \\ & > & \lambda c^T x + (1-\lambda) c^T x \\ & = & c^T x \end{array}$$

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda)z$.
- Es folgt:

$$\begin{aligned} c^T y &= c^T(\lambda x + (1-\lambda)z) \\ &= \lambda c^T x + (1-\lambda)c^T z \\ &> \lambda c^T x + (1-\lambda)c^T x \\ &= c^T x \end{aligned}$$

Lemma

Ein lokales Optimum ist auch ein globales Optimum.

Lokales und globales Optimum

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda) z$.
- Es folgt:

$$\begin{aligned} c^T y &= c^T(\lambda x + (1-\lambda) z) \\ &= \lambda c^T x + (1-\lambda) c^T z \\ &> \lambda c^T x + (1-\lambda) c^T x \\ &= c^T x \end{aligned}$$

Lemma

Ein lokales Optimum ist auch ein globales Optimum.

Unterräume

- Eine Hyperebene wird beschrieben durch:

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

Unterräume

P **Polyhedron**

- Eine Hyperebene wird beschrieben durch:

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d-1$ Variablen frei wählbar.

Unterräume

P **Polyhedron**

- Eine Hyperebene wird beschrieben durch:

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d - 1$ Variablen frei wählbar.
  - Der Wert der d-ten Variable ist dann festgelegt.

Unterräume

- Eine Hyperebene wird beschrieben durch:

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d-1$ Variablen frei wählbar.
  - Der Wert der d-ten Variable ist dann festgelegt.
  - Der durch die Hyperebene beschriebene affine Unterraum ist ein Unterraum der Dimension $d-1$.

Unterräume

- Eine Hyperebene wird beschrieben durch:

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d-1$ Variablen frei wählbar.
  - Der Wert der d-ten Variable ist dann festgelegt.
  - Der durch die Hyperebene beschriebene affine Unterraum ist ein Unterraum der Dimension $d-1$.
  - Ein Unterraum, der als Schnittmenge von k linear unabhängigen Hyperräumen beschrieben wird, hat Dimension $d-k$.

Unterräume

- Eine Hyperebene wird beschrieben durch:

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d-1$ Variablen frei wählbar.
  - Der Wert der d-ten Variable ist dann festgelegt.
  - Der durch die Hyperebene beschriebene affine Unterraum ist ein Unterraum der Dimension $d-1$.
  - Ein Unterraum, der als Schnittmenge von k linear unabhängigen Hyperräumen beschrieben wird, hat Dimension $d-k$.
- Falls mehr als d Nebenbedingungen (Hyperebenen) sich in einem Punkt treffen, so ist das LP degeneriert.

Unterräume

- Eine Hyperebene wird beschrieben durch:

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d-1$ Variablen frei wählbar.
  - Der Wert der d-ten Variable ist dann festgelegt.
  - Der durch die Hyperebene beschriebene affine Unterraum ist ein Unterraum der Dimension $d-1$.
  - Ein Unterraum, der als Schnittmenge von k linear unabhängigen Hyperräumen beschrieben wird, hat Dimension $d-k$.
- Falls mehr als d Nebenbedingungen (Hyperebenen) sich in einem Punkt treffen, so ist das LP degeneriert.
- Ein degeneriertes LP kann in ein nicht-degeneriertes LP umgeformt werden, ohne die Form (Zusammensetzung) der Lösung signifikant zu verändern.

Unterräume

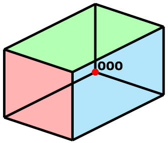
- Eine Hyperebene wird beschrieben durch:

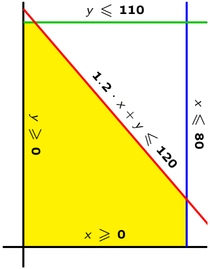
$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d-1$ Variablen frei wählbar.
  - Der Wert der d-ten Variable ist dann festgelegt.
  - Der durch die Hyperebene beschriebene affine Unterraum ist ein Unterraum der Dimension $d-1$.
  - Ein Unterraum, der als Schnittmenge von k linear unabhängigen Hyperräumen beschrieben wird, hat Dimension $d-k$.
- Falls mehr als d Nebenbedingungen (Hyperebenen) sich in einem Punkt treffen, so ist das LP degeneriert.
- Ein degeneriertes LP kann in ein nicht-degeneriertes LP umgeformt werden, ohne die Form (Zusammensetzung) der Lösung signifikant zu verändern.

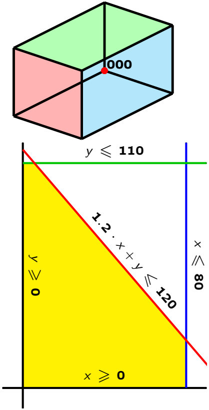
Oberfläche eines Polyhedrons

P **Polyhedron**





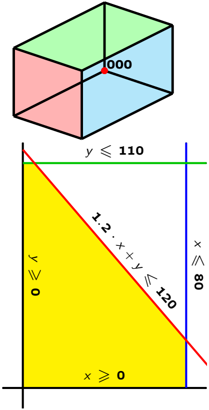
Oberfläche eines Polyhedrons



- Die Oberfläche eines Polyhedrons besteht aus Facetten.

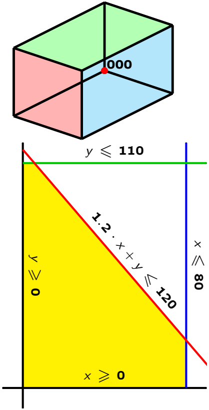
Oberfläche eines Polyhedrons

P **Polyhedron**



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.

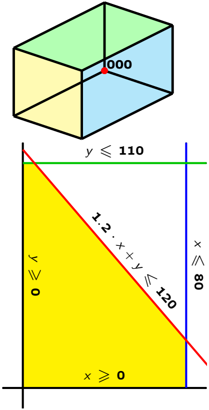
Oberfläche eines Polyhedrons



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so

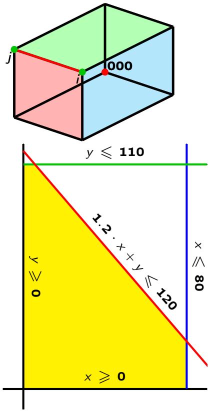
Oberfläche eines Polyhedrons

P **Polyhedron**



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.

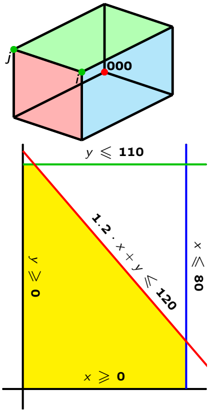
Oberfläche eines Polyhedrons



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d - 1$ heißt Face.

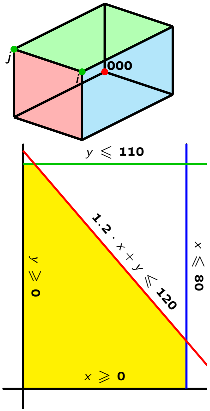
Oberfläche eines Polyhedrons

P **Polyhedron**



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d - 1$ heißt Face.
- Eine Kante entspricht dem Schnitt von $d - 1$ Hyperebenen (Facette der Dimension 1).

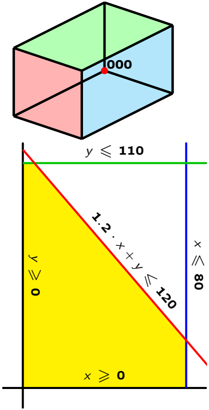
Oberfläche eines Polyhedrons



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d-1$ heißt Face.
- Eine Kante entspricht dem Schnitt von $d-1$ Hyperebenen (Facette der Dimension 1).
- Ein Knoten entspricht dem Schnitt von d Hyperebenen (Facette der Dimension 0).

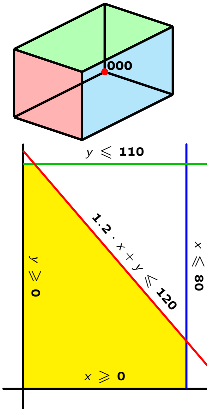
Oberfläche eines Polyhedrons

P **Polyhedron**



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d - 1$ heißt Face.
- Eine Kante entspricht dem Schnitt von $d - 1$ Hyperebenen (Facette der Dimension 1).
- Ein Knoten entspricht dem Schnitt von d Hyperebenen (Facette der Dimension 0).
- Zwei Knoten sind benachbart, wenn sie durch eine Kante verbunden sind.

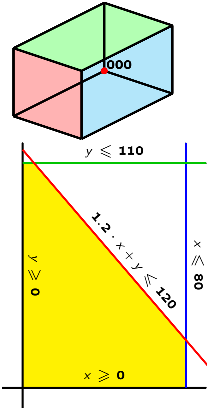
Oberfläche eines Polyhedrons



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d - 1$ heißt Face.
- Eine Kante entspricht dem Schnitt von $d - 1$ Hyperebenen (Facette der Dimension 1).
- Ein Knoten entspricht dem Schnitt von d Hyperebenen (Facette der Dimension 0).
- Zwei Knoten sind benachbart, wenn sie durch eine Kante verbunden sind.
- Falls P unbeschränkt ist, so kann es unbeschränkte Kanten geben.

Oberfläche eines Polyhedrons

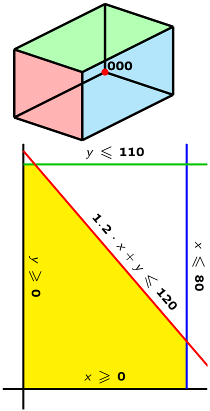
P **Polyhedron**



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d - 1$ heißt Face.
- Eine Kante entspricht dem Schnitt von $d - 1$ Hyperebenen (Facette der Dimension 1).
- Ein Knoten entspricht dem Schnitt von d Hyperebenen (Facette der Dimension 0).
- Zwei Knoten sind benachbart, wenn sie durch eine Kante verbunden sind.
- Falls P unbeschränkt ist, so kann es unbeschränkte Kanten geben.
- Solche Kanten haben nur einen oder keinen Endpunkt.

Oberfläche eines Polyhedrons

P **Polyhedron**



- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d-1$ heißt Face.
- Eine Kante entspricht dem Schnitt von $d-1$ Hyperebenen (Facette der Dimension 1).
- Ein Knoten entspricht dem Schnitt von d Hyperebenen (Facette der Dimension 0).
- Zwei Knoten sind benachbart, wenn sie durch eine Kante verbunden sind.
- Falls P unbeschränkt ist, so kann es unbeschränkte Kanten geben.
- Solche Kanten haben nur einen oder keinen Endpunkt.

Zielfunktion

- Die Zielfunktion $c^T x$ (Vektor) gibt eine Richtung in $\mathbb{R}^d$ vor.

Zielfunktion

- Die Zielfunktion $c^T x$ (Vektor) gibt eine Richtung in $\mathbb{R}^d$ vor.
- Falls die Nebenbedingungen den Zielwert nach oben beschränken, so wird das LP als beschränkt bezeichnet.

Zielfunktion

- Die Zielfunktion $c^T x$ (Vektor) gibt eine Richtung in $\mathbb{R}^d$ vor.
- Falls die Nebenbedingungen den Zielwert nach oben beschränken, so wird das LP als beschränkt bezeichnet.
  - Falls der Zielwert nicht beschränkt ist, so heißt das LP unbeschränkt.

Zielfunktion

- Die Zielfunktion $c^T x$ (Vektor) gibt eine Richtung in $\mathbb{R}^d$ vor.
- Falls die Nebenbedingungen den Zielwert nach oben beschränken, so wird das LP als beschränkt bezeichnet.
  - Falls der Zielwert nicht beschränkt ist, so heißt das LP unbeschränkt.
- Das Polyhedron muss dabei nur in der Richtung von $c^T x$ beschränkt sein.

Zielfunktion

- Die Zielfunktion $c^T x$ (Vektor) gibt eine Richtung in $\mathbb{R}^d$ vor.
- Falls die Nebenbedingungen den Zielwert nach oben beschränken, so wird das LP als beschränkt bezeichnet.
  - Falls der Zielwert nicht beschränkt ist, so heißt das LP unbeschränkt.
- Das Polyhedron muss dabei nur in der Richtung von $c^T x$ beschränkt sein.
- Ist das Polyhedron in alle Richtungen beschränkt (in einer Kugel enthalten), so wird es als Polytop bezeichnet.

Zielfunktion

- Die Zielfunktion $c^T x$ (Vektor) gibt eine Richtung in $\mathbb{R}^d$ vor.
- Falls die Nebenbedingungen den Zielwert nach oben beschränken, so wird das LP als beschränkt bezeichnet.
  - Falls der Zielwert nicht beschränkt ist, so heißt das LP unbeschränkt.
- Das Polyhedron muss dabei nur in der Richtung von $c^T x$ beschränkt sein.
- Ist das Polyhedron in alle Richtungen beschränkt (in einer Kugel enthalten), so wird es als Polytop bezeichnet.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.
- Wähle z maximal mit $P \cap \mathcal{H}_z \neq \emptyset$.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.
- Wähle z maximal mit $P \cap \mathcal{H}_z \neq \emptyset$.
- Ein beliebiger Punkt $x^* \in P \cap \mathcal{H}_z$ ist eine optimale Lösung des LPs.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.
- Wähle z maximal mit $P \cap \mathcal{H}_z \neq \emptyset$.
- Ein beliebiger Punkt $x^* \in P \cap \mathcal{H}_z$ ist eine optimale Lösung des LPs.
- Beobachtungen:

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.
- Wähle z maximal mit $P \cap \mathcal{H}_z \neq \emptyset$.
- Ein beliebiger Punkt $x^* \in P \cap \mathcal{H}_z$ ist eine optimale Lösung des LPs.
- Beobachtungen:
    - $P \cap \mathcal{H}_z$ ist eine Facette f von P.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.
- Wähle z maximal mit $P \cap \mathcal{H}_z \neq \emptyset$.
- Ein beliebiger Punkt $x^* \in P \cap \mathcal{H}_z$ ist eine optimale Lösung des LPs.
- Beobachtungen:
  - $P \cap \mathcal{H}_z$ ist eine Facette f von P.
  - Falls f nicht in allen Richtungen unbeschränkt ist, so gibt es mindestens einen optimalen Knoten.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.
- Wähle z maximal mit $P \cap \mathcal{H}_z \neq \emptyset$.
- Ein beliebiger Punkt $x^* \in P \cap \mathcal{H}_z$ ist eine optimale Lösung des LPs.
- Beobachtungen:
  - $P \cap \mathcal{H}_z$ ist eine Facette f von P.
  - Falls f nicht in allen Richtungen unbeschränkt ist, so gibt es mindestens einen optimalen Knoten.

Die Idee

- Kann eine zufällige Auswahl helfen?

Die Idee

- Kann eine zufällige Auswahl helfen?
- Können effiziente Algorithmen noch schneller gemacht werden?

Die Idee

- Kann eine zufällige Auswahl helfen?
- Können effiziente Algorithmen noch schneller gemacht werden?
- Kann bei schweren Problemen mit ausreichender Wahrscheinlichkeit ein gutes Ergebnis erreicht werden?

Die Idee

- Kann eine zufällige Auswahl helfen?
- Können effiziente Algorithmen noch schneller gemacht werden?
- Kann bei schweren Problemen mit ausreichender Wahrscheinlichkeit ein gutes Ergebnis erreicht werden?

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:
  - Maximiere f unter Berücksichtigung aller Nebenbedingungen.

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:
  - Maximiere f unter Berücksichtigung aller Nebenbedingungen.
- Das ist ein LP (lineares Programm)

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:
  - Maximiere f unter Berücksichtigung aller Nebenbedingungen.
- Das ist ein LP (lineares Programm)
- Obere Schranke für Laufzeit: $O(\binom{m}{d}) = O(n^d)$.

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:
  - Maximiere f unter Berücksichtigung aller Nebenbedingungen.
- Das ist ein LP (lineares Programm)
- Obere Schranke für Laufzeit: $O(\binom{m}{d}) = O(n^d)$.
  - Untersuche für jede d-elementige Teilmenge der m Ungleichungen den jeweiligen Schnittpunkt (Basislösung).

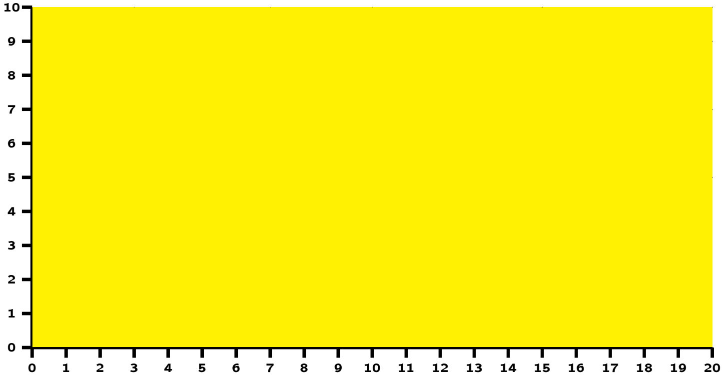
Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:
  - Maximiere f unter Berücksichtigung aller Nebenbedingungen.
- Das ist ein LP (lineares Programm)
- Obere Schranke für Laufzeit: $O(\binom{m}{d}) = O(n^d)$.
  - Untersuche für jede d-elementige Teilmenge der m Ungleichungen den jeweiligen Schnittpunkt (Basislösung).
- Hier nun Algorithmus mit erwarteter linearer Laufzeit (d.h. linear in m).

Einleitung

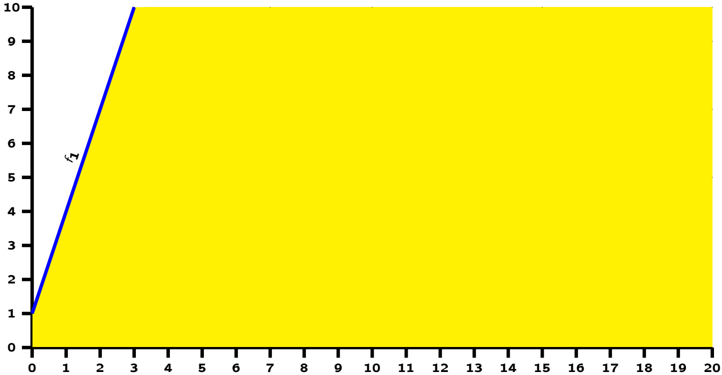
- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:
  - Maximiere f unter Berücksichtigung aller Nebenbedingungen.
- Das ist ein LP (lineares Programm)
- Obere Schranke für Laufzeit: $O(\binom{m}{d}) = O(n^d)$.
  - Untersuche für jede d-elementige Teilmenge der m Ungleichungen den jeweiligen Schnittpunkt (Basislösung).
- Hier nun Algorithmus mit erwarteter linearer Laufzeit (d.h. linear in m).

Beispiel



Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

Beispiel

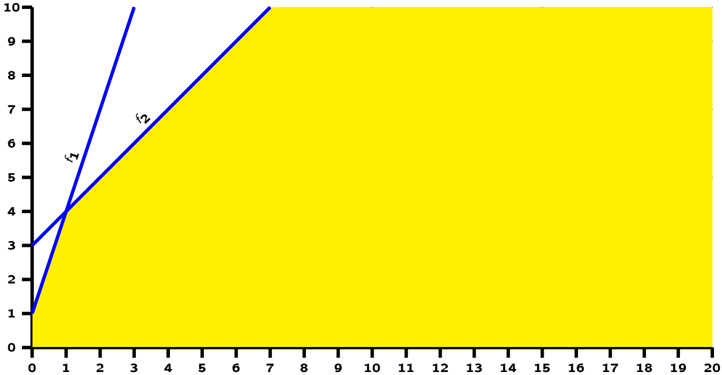


Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$f_1 : y \leqslant 3 \cdot x + 1$$

Beispiel

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

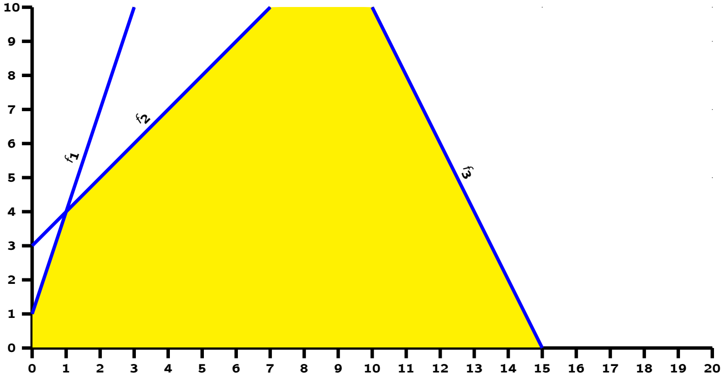


Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$f_1 : y \leqslant 3 \cdot x + 1$$
$$f_2 : y \leqslant 1 \cdot x + 2$$

Beispiel

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

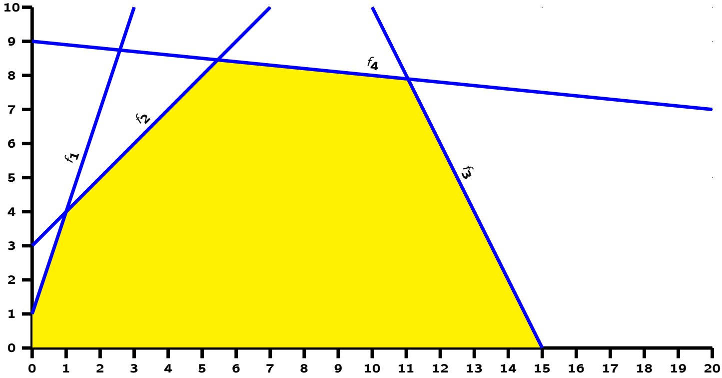


Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$f_1 : y \leqslant 3 \cdot x + 1$$
$$f_2 : y \leqslant 1 \cdot x + 2$$
$$f_3 : y \leqslant -2 \cdot x + 30$$

Beispiel

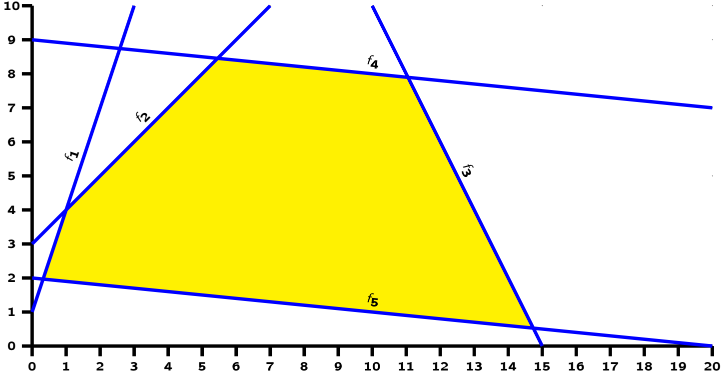
d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n



Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$f_1 : y \leqslant 3 \cdot x + 1 \qquad f_4 : y \leqslant -0.9 \cdot x + 9$$
$$f_2 : y \leqslant 1 \cdot x + 2$$
$$f_3 : y \leqslant -2 \cdot x + 30$$

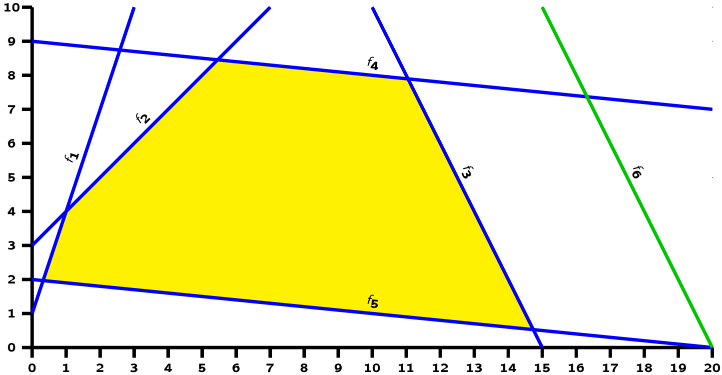
Beispiel



Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$\begin{aligned} f_1 &: y \leqslant 3 \cdot x + 1 & f_4 &: y \leqslant -0.9 \cdot x + 9 \\ f_2 &: y \leqslant 1 \cdot x + 2 & f_5 &: y \geqslant -0.9 \cdot x + 2 \\ f_3 &: y \leqslant -2 \cdot x + 30 & & \end{aligned}$$

Beispiel

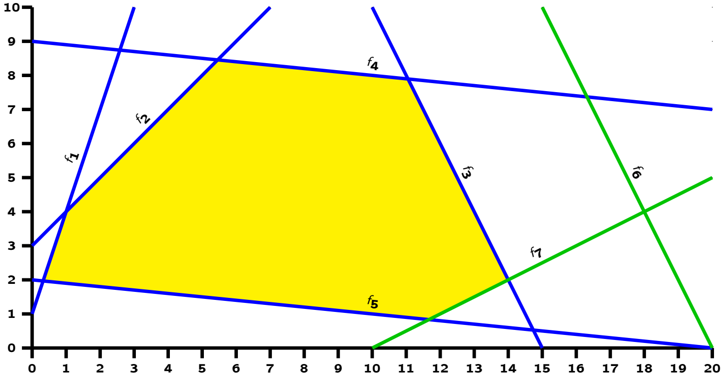


Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$\begin{array}{ll} f_1 : y \leqslant 3 \cdot x + 1 & f_4 : y \leqslant -0.9 \cdot x + 9 \\ f_2 : y \leqslant 1 \cdot x + 2 & f_5 : y \geqslant -0.9 \cdot x + 2 \\ f_3 : y \leqslant -2 \cdot x + 30 & f_6 : y \leqslant -2 \cdot x + 40 \end{array}$$

Beispiel

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

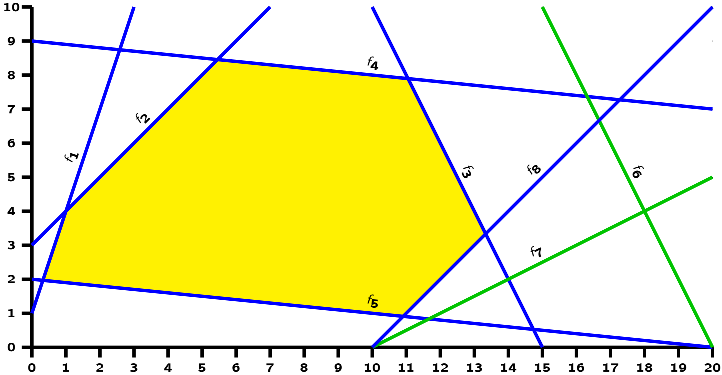


Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$\begin{array}{lll}
f_1 : y \leqslant 3 \cdot x + 1 & f_4 : y \leqslant -0.9 \cdot x + 9 & f_7 : y \geqslant 0.5 \cdot x - 5 \\
f_2 : y \leqslant 1 \cdot x + 2 & f_5 : y \geqslant -0.9 \cdot x + 2 & \\
f_3 : y \leqslant -2 \cdot x + 30 & f_6 : y \leqslant -2 \cdot x + 40 &
\end{array}$$

Beispiel

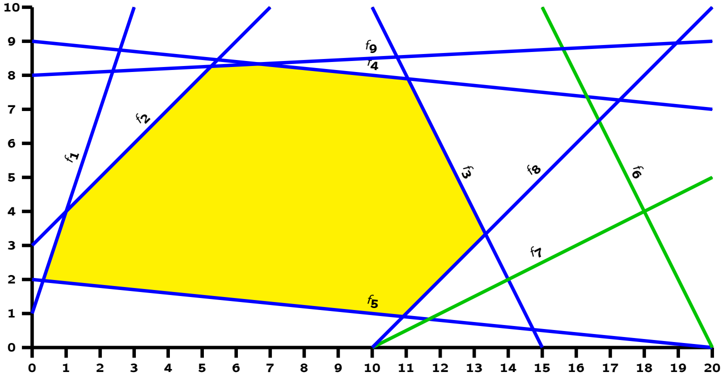
d **Variablen,** m **Nebenbedingungen, Dimentsion** n



Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$
\begin{array}{lll}
f_1 : y \leqslant 3 \cdot x + 1 & f_4 : y \leqslant -0.9 \cdot x + 9 & f_7 : y \geqslant 0.5 \cdot x - 5 \\
f_2 : y \leqslant 1 \cdot x + 2 & f_5 : y \geqslant -0.9 \cdot x + 2 & f_8 : y \geqslant 1 \cdot x - 10 \\
f_3 : y \leqslant -2 \cdot x + 30 & f_6 : y \leqslant -2 \cdot x + 40 &
\end{array}
$$

Beispiel



Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$$
\begin{array}{lll}
f_1 : y \leqslant 3 \cdot x + 1 & f_4 : y \leqslant -0.9 \cdot x + 9 & f_7 : y \geqslant 0.5 \cdot x - 5 \\
f_2 : y \leqslant 1 \cdot x + 2 & f_5 : y \geqslant -0.9 \cdot x + 2 & f_8 : y \geqslant 1 \cdot x - 10 \\
f_3 : y \leqslant -2 \cdot x + 30 & f_6 : y \leqslant -2 \cdot x + 40 & f_9 : y \leqslant 1.05 \cdot x + 8
\end{array}
$$

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.
  - Falls nicht, so setze:

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.
  - Falls nicht, so setze:
  - $c_i = c_i + \varepsilon^i$ für ein $\varepsilon > 0$.

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.
  - Falls nicht, so setze:
  - $c_i = c_i + \varepsilon^i$ für ein $\varepsilon > 0$.
  - Zielfunktion wird virtuell perturbiert (durcheinander wirbeln,stören).

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.
  - Falls nicht, so setze:
  - $c_i = c_i + \varepsilon^i$ für ein $\varepsilon > 0$.
  - Zielfunktion wird virtuell perturbiert (durcheinander wirbeln,stören).
  - D.h. es wird lexikographisch kleinste Basislösung gewählt.

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.
  - Falls nicht, so setze:
  - $c_i = c_i + \varepsilon^i$ für ein $\varepsilon > 0$.
  - Zielfunktion wird virtuell perturbiert (durcheinander wirbeln,stören).
  - D.h. es wird lexikographisch kleinste Basislösung gewählt.
- Jede Nebenbedingung entspricht einer Hyperebene.

Vorgaben

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.
  - Falls nicht, so setze:
  - $c_i = c_i + \varepsilon^i$ für ein $\varepsilon > 0$.
  - Zielfunktion wird virtuell perturbiert (durcheinander wirbeln,stören).
  - D.h. es wird lexikographisch kleinste Basislösung gewählt.
- Jede Nebenbedingung entspricht einer Hyperebene.

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Das LP sollte nicht unbeschränkt sein.

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.

Vorgaben

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
    - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
    - t muss so groß sein, dass keine Basislösung verloren geht.

Vorgaben

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
  - t muss so groß sein, dass keine Basislösung verloren geht.
  - D.h. in jeder Basislösung müssen die Variablenwerte zwischen t und $-t$ liegen.

Vorgaben

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
  - t muss so groß sein, dass keine Basislösung verloren geht.
  - D.h. in jeder Basislösung müssen die Variablenwerte zwischen t und $-t$ liegen.
  - So ein Wert t existiert immer.

Vorgaben

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
  - t muss so groß sein, dass keine Basislösung verloren geht.
  - D.h. in jeder Basislösung müssen die Variablenwerte zwischen t und $-t$ liegen.
  - So ein Wert t existiert immer.
  - t kann polynomiell in der Eingabelänge bestimmt werden.

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
  - t muss so groß sein, dass keine Basislösung verloren geht.
  - D.h. in jeder Basislösung müssen die Variablenwerte zwischen t und $-t$ liegen.
  - So ein Wert t existiert immer.
  - t kann polynomiell in der Eingabelänge bestimmt werden.
  - Man kann auch t als symbolischen Wert darstellen.

Vorgaben

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
  - t muss so groß sein, dass keine Basislösung verloren geht.
  - D.h. in jeder Basislösung müssen die Variablenwerte zwischen t und $-t$ liegen.
  - So ein Wert t existiert immer.
  - t kann polynomiell in der Eingabelänge bestimmt werden.
  - Man kann auch t als symbolischen Wert darstellen.
  - Dann wird t im folgenden Algorithmus immer größer sein als jeder bis dahin berechnete Wert.

Vorgaben

- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
  - t muss so groß sein, dass keine Basislösung verloren geht.
  - D.h. in jeder Basislösung müssen die Variablenwerte zwischen t und $-t$ liegen.
  - So ein Wert t existiert immer.
  - t kann polynomiell in der Eingabelänge bestimmt werden.
  - Man kann auch t als symbolischen Wert darstellen.
  - Dann wird t im folgenden Algorithmus immer größer sein als jeder bis dahin berechnete Wert.

Algorithmus von Seidel (1991)

- Idee:

Algorithmus von Seidel (1991)

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und

Algorithmus von Seidel (1991)

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.
- Algorithmus von Seidel:

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.
- Algorithmus von Seidel:
  1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.
- Algorithmus von Seidel:
  1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
  2. Ansonsten wähle uniform eine Nebenbedingung $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.
- Algorithmus von Seidel:
  1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
  2. Ansonsten wähle uniform eine Nebenbedingung $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
  3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.

Algorithmus von Seidel (1991)

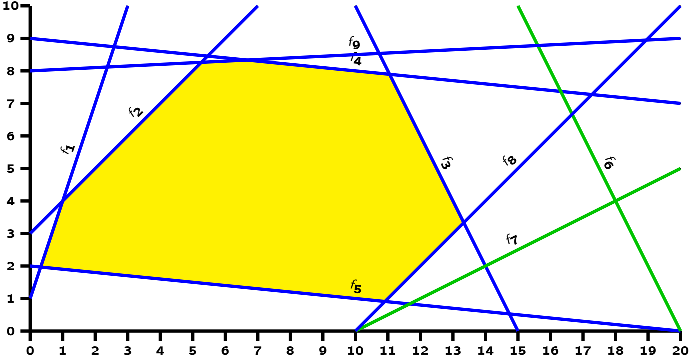
d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.
- Algorithmus von Seidel:
  1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
  2. Ansonsten wähle uniform eine Nebenbedingung $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
  3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
  4. Ansonsten berechne den Schnitt des Lösungspolyhedrons mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Algorithmus von Seidel (1991)

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.
- Algorithmus von Seidel:
  1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
  2. Ansonsten wähle uniform eine Nebenbedingung $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
  3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
  4. Ansonsten berechne den Schnitt des Lösungspolyhedrons mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)



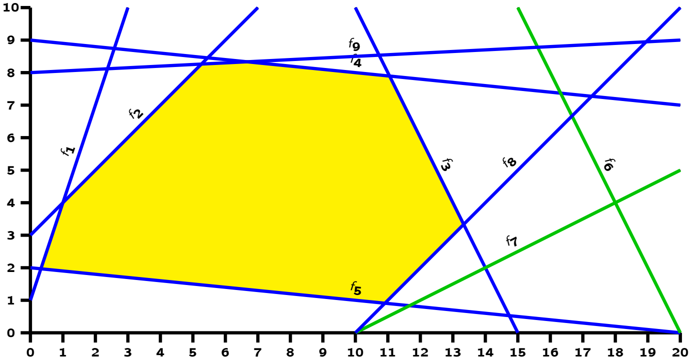
1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Entferne nacheinander

$$f_1 : y \leqslant 3 \cdot x + 1 \qquad f_4 : y \leqslant -0.9 \cdot x + 9 \qquad f_7 : y \geqslant 0.5 \cdot x - 5$$
$$f_2 : y \leqslant 1 \cdot x + 2 \qquad f_5 : y \geqslant -0.9 \cdot x + 2 \qquad f_8 : y \geqslant 1 \cdot x - 10$$
$$f_3 : y \leqslant -2 \cdot x + 30 \qquad f_6 : y \leqslant -2 \cdot x + 40 \qquad f_9 : y \leqslant 1.05 \cdot x + 8$$

Beispiel (Abstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Entferne nacheinander f_8,

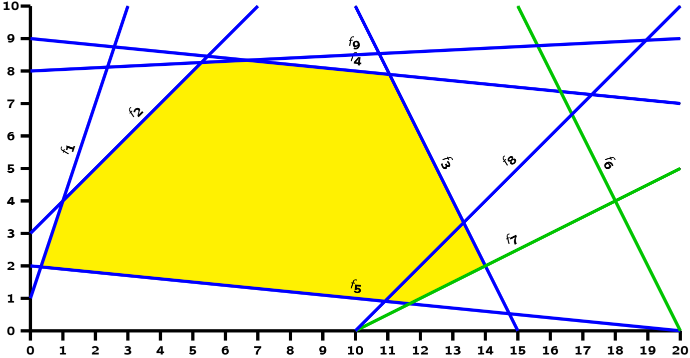
$f_1 : y \leqslant 3 \cdot x + 1$ $\quad f_4 : y \leqslant -0.9 \cdot x + 9$ $\quad f_7 : y \geqslant 0.5 \cdot x - 5$

$f_2 : y \leqslant 1 \cdot x + 2$ $\quad f_5 : y \geqslant -0.9 \cdot x + 2$ $\quad f_8 : y \geqslant 1 \cdot x - 10$

$f_3 : y \leqslant -2 \cdot x + 30$ $\quad f_6 : y \leqslant -2 \cdot x + 40$ $\quad f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)



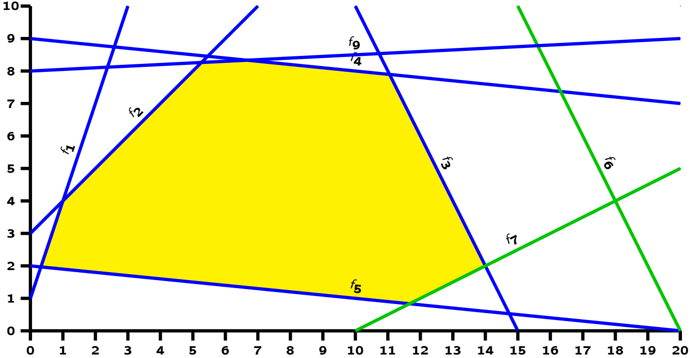
Entferne nacheinander f_8,

$$f_1: y \leqslant 3 \cdot x + 1 \quad f_4: y \leqslant -0.9 \cdot x + 9 \quad f_7: y \geqslant 0.5 \cdot x - 5$$
$$f_2: y \leqslant 1 \cdot x + 2 \quad f_5: y \geqslant -0.9 \cdot x + 2 \quad f_8: y \geqslant 1 \cdot x - 10$$
$$f_3: y \leqslant -2 \cdot x + 30 \quad f_6: y \leqslant -2 \cdot x + 40 \quad f_9: y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n

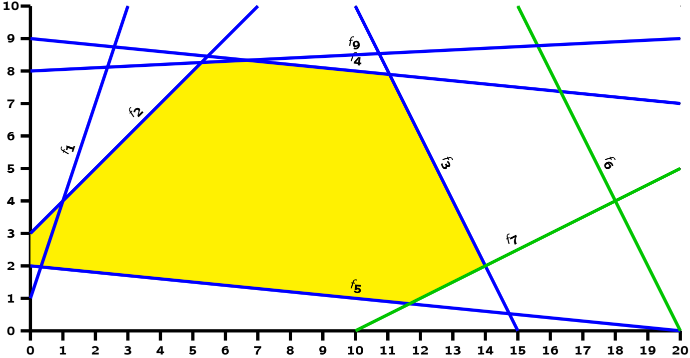


Entferne nacheinander f_8, f_1,

$$f_1 : y \leqslant 3 \cdot x + 1 \qquad f_4 : y \leqslant -0.9 \cdot x + 9 \qquad f_7 : y \geqslant 0.5 \cdot x - 5$$
$$f_2 : y \leqslant 1 \cdot x + 2 \qquad f_5 : y \geqslant -0.9 \cdot x + 2$$
$$f_3 : y \leqslant -2 \cdot x + 30 \qquad f_6 : y \leqslant -2 \cdot x + 40 \qquad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)



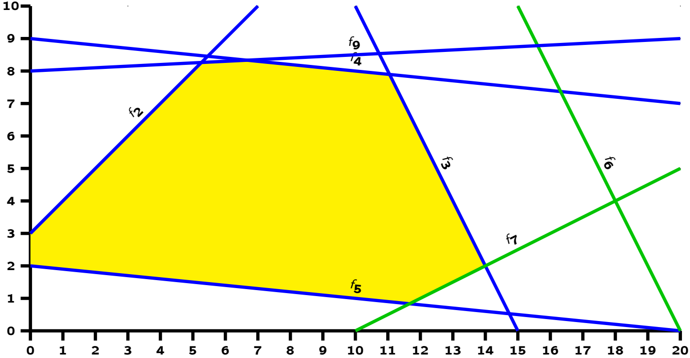
Entferne nacheinander f_8, f_1,

$f_1 : y \leqslant 3 \cdot x + 1$	$f_4 : y \leqslant -0.9 \cdot x + 9$	$f_7 : y \geqslant 0.5 \cdot x - 5$
$f_2 : y \leqslant 1 \cdot x + 2$	$f_5 : y \geqslant -0.9 \cdot x + 2$	
$f_3 : y \leqslant -2 \cdot x + 30$	$f_6 : y \leqslant -2 \cdot x + 40$	$f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



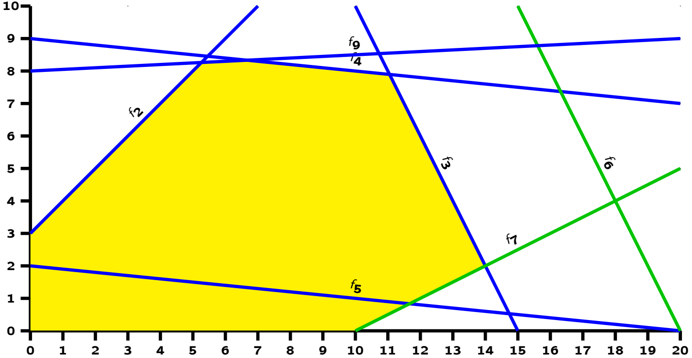
Entferne nacheinander f_8, f_1, f_5

$$\begin{array}{lll} & f_4 : y \leqslant -0.9 \cdot x + 9 & f_7 : y \geqslant 0.5 \cdot x - 5 \\ f_2 : y \leqslant 1 \cdot x + 2 & f_5 : y \geqslant -0.9 \cdot x + 2 & \\ f_3 : y \leqslant -2 \cdot x + 30 & f_6 : y \leqslant -2 \cdot x + 40 & f_9 : y \leqslant 1.05 \cdot x + 8 \end{array}$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



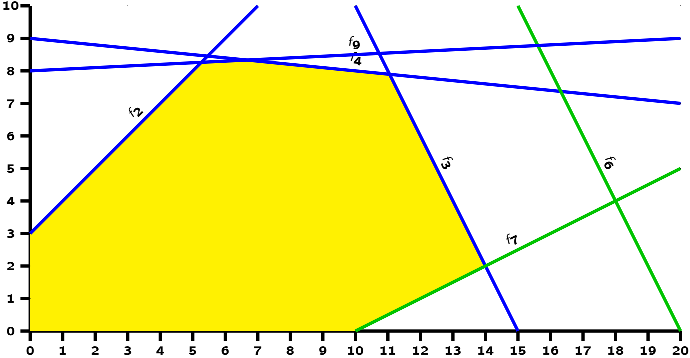
Entferne nacheinander f_8, f_1, f_5

$$f_4 : y \leqslant -0.9 \cdot x + 9 \quad f_7 : y \geqslant 0.5 \cdot x - 5$$
$$f_2 : y \leqslant 1 \cdot x + 2 \quad f_5 : y \geqslant -0.9 \cdot x + 2$$
$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Entferne nacheinander f_8, f_1, f_5 und f_7 und löse jeweils rekursiv.

$f_4 : y \leqslant -0.9 \cdot x + 9$ $f_7 : y \geqslant 0.5 \cdot x - 5$

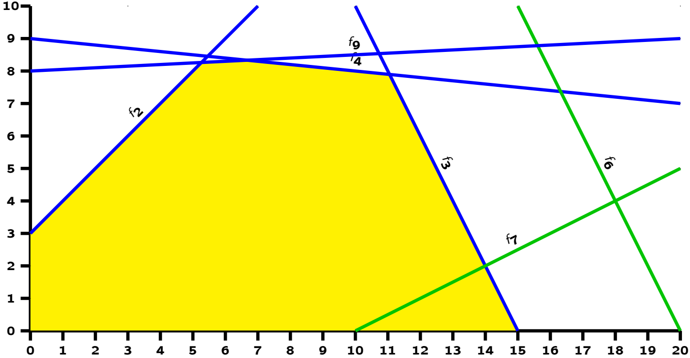
$f_2 : y \leqslant 1 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30$ $f_6 : y \leqslant -2 \cdot x + 40$ $f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n



Entferne nacheinander f_8, f_1, f_5 und f_7 und löse jeweils rekursiv.

$f_4 : y \leqslant -0.9 \cdot x + 9 \quad f_7 : y \geqslant 0.5 \cdot x - 5$

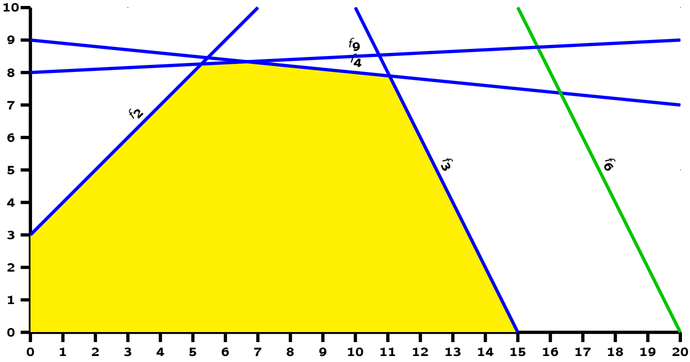
$f_2 : y \leqslant 1 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ **oder** $m = 0$, **so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h, **und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Beispiel (Abstieg der nächsten vier Rekursionen)

d **Variablen,** m **Nebenbedingungen, Dimentsion** n



1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3.Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Entferne nacheinander

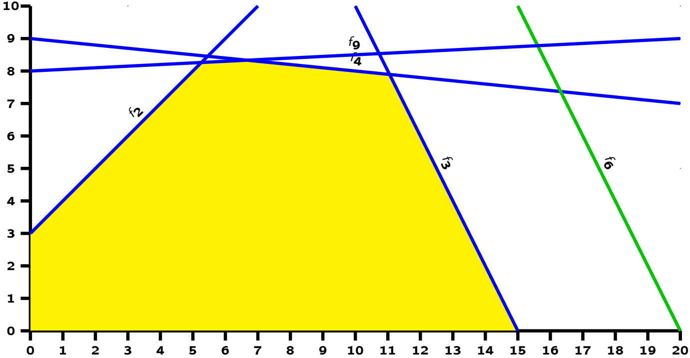
$f_4 : y \leqslant -0.9 \cdot x + 9$

$f_2 : y \leqslant 1 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30$ $\quad f_6 : y \leqslant -2 \cdot x + 40$ $\quad f_9 : y \leqslant 1.05 \cdot x + 8$

Beispiel (Abstieg der nächsten vier Rekursionen)

d **Variablen,** m **Nebenbedingungen, Dimentsion** n



Entferne nacheinander f_4,

$$f_4 : y \leqslant -0.9 \cdot x + 9$$

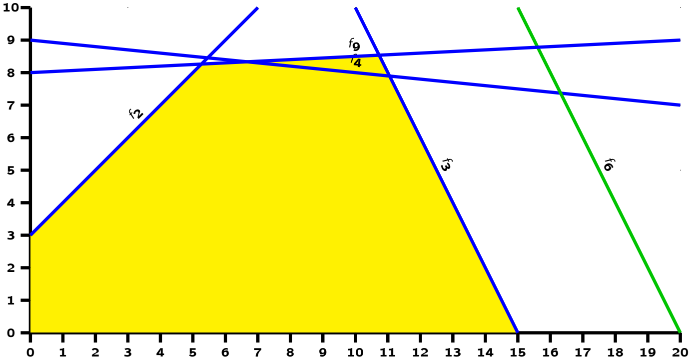
$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der nächsten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Entferne nacheinander f_4,

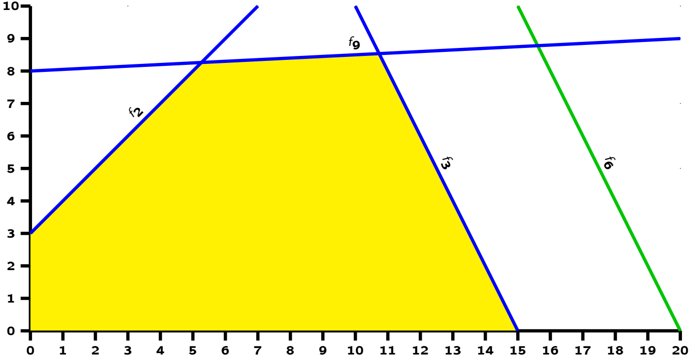
$$f_4 : y \leqslant -0.9 \cdot x + 9$$

$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \qquad f_6 : y \leqslant -2 \cdot x + 40 \qquad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der nächsten vier Rekursionen)



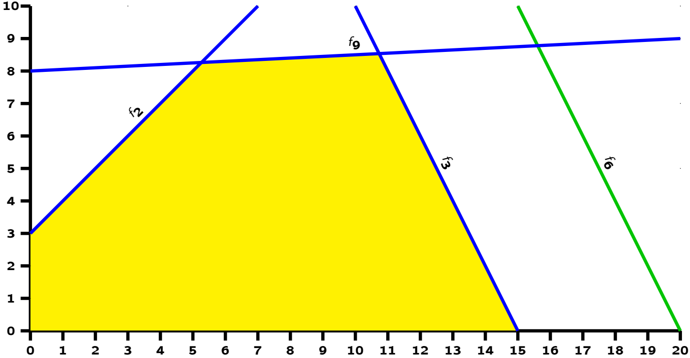
Entferne nacheinander f_4, f_6,

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

$$f_2 : y \leqslant 1 \cdot x + 2$$
$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

Beispiel (Abstieg der nächsten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Entferne nacheinander f_4, f_6,

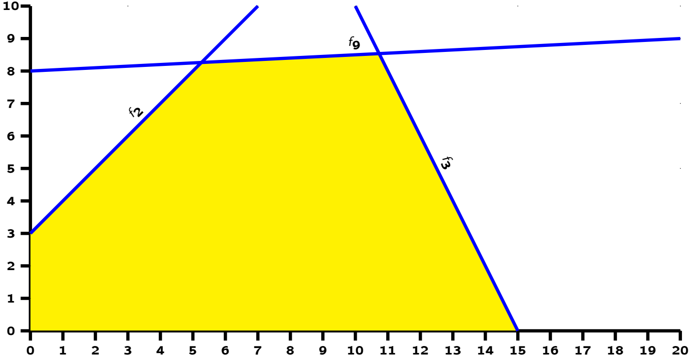
1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

$f_2 : y \leqslant 1 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$

Beispiel (Abstieg der nächsten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Entferne nacheinander f_4, f_6, f_3

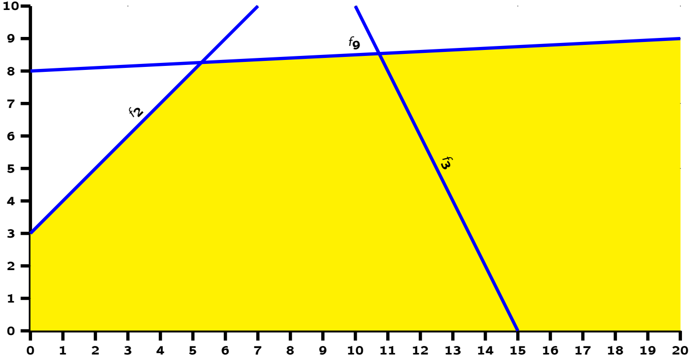
$f_2 : y \leqslant 1 \cdot x + 2$
$f_3 : y \leqslant -2 \cdot x + 30$

$f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der nächsten vier Rekursionen)

d **Variablen,** m **Nebenbedingungen, Dimentsion** n



Entferne nacheinander f_4, f_6, f_3

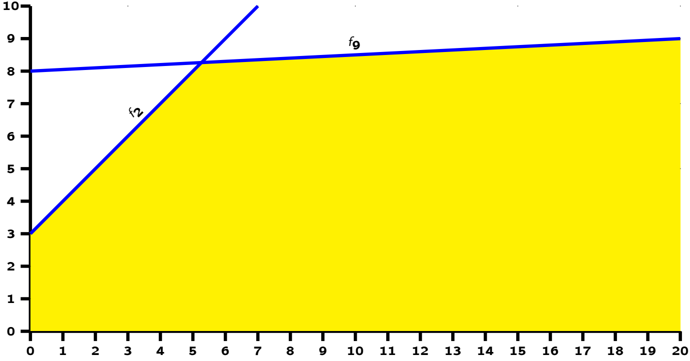
$f_2 : y \leqslant 1 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30$

$f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3.Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Beispiel (Abstieg der nächsten vier Rekursionen)



Entferne nacheinander f_4, f_6, f_3 und f_2 und löse jeweils rekursiv.

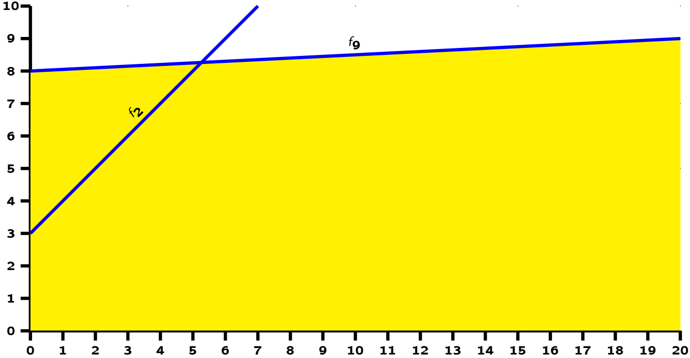
$f_2 : y \leqslant 1 \cdot x + 2$

$f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der nächsten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Entferne nacheinander f_4, f_6, f_3 und f_2 und löse jeweils rekursiv.

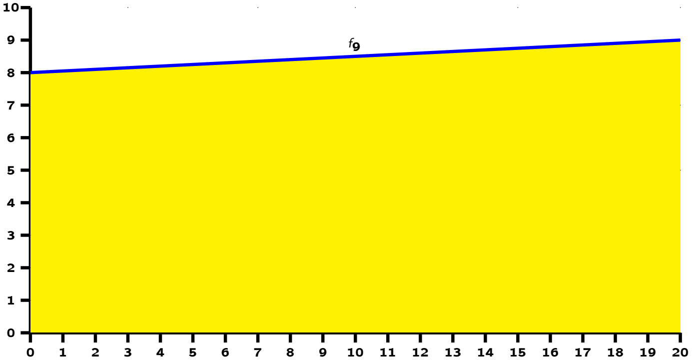
1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_9 : y \leqslant 1.05 \cdot x + 8$$

Beispiel (Abstieg der nächsten vier Rekursionen)

d **Variablen,** m **Nebenbedingungen, Dimentsion** n

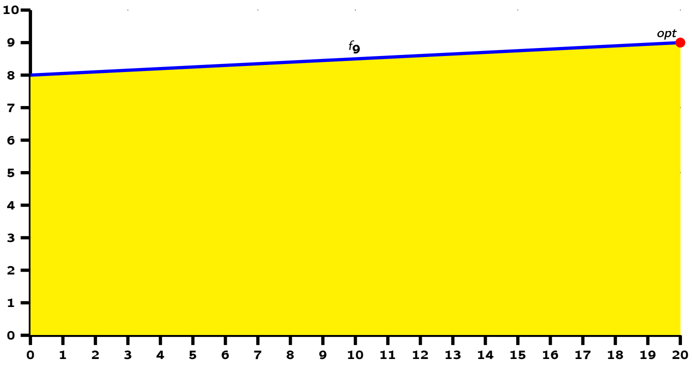


Entferne nacheinander f_4, f_6, f_3 und f_2 und löse jeweils rekursiv.

$$f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3.Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Beispiel (Abstieg der nächsten vier Rekursionen)



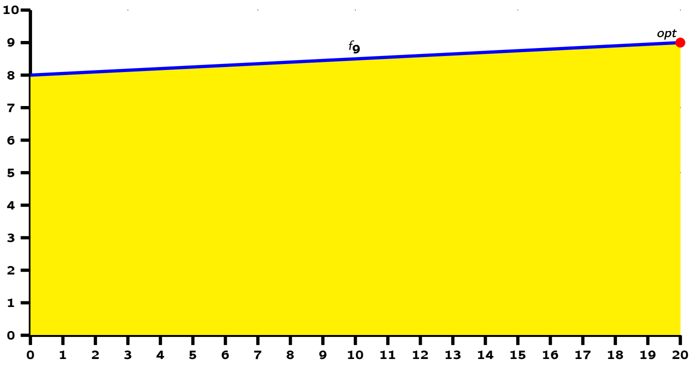
Entferne nacheinander f_4, f_6, f_3 und f_2 und löse jeweils rekursiv.

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3.Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

$$f_9 : y \leqslant 1.05 \cdot x + 8$$

Beispiel (Aufstieg der letzten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



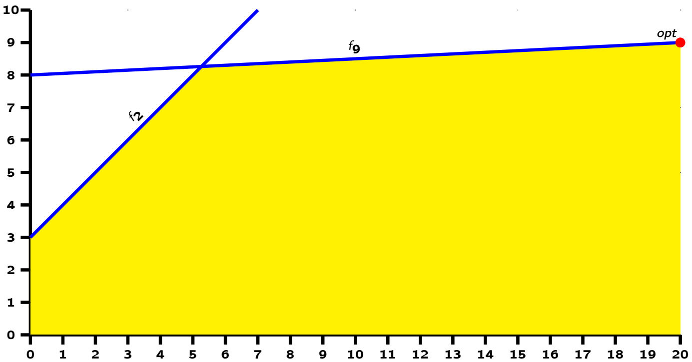
Füge nacheinander wieder ein:

$$f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der letzten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Füge nacheinander wieder ein: f_2,

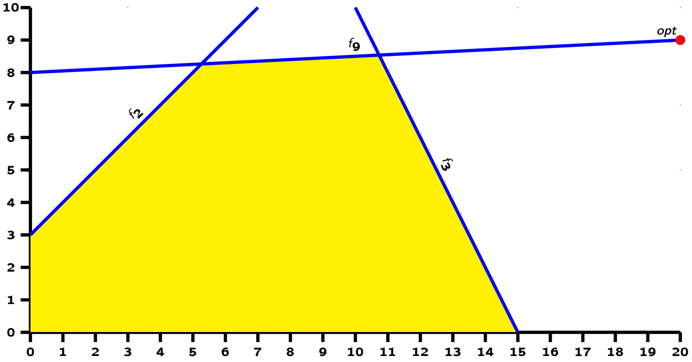
$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der letzten vier Rekursionen)

d **Variablen,** m **Nebenbedingungen, Dimentsion** n



Füge nacheinander wieder ein: f_2, f_3,

$f_2 : y \leqslant 1 \cdot x + 2$

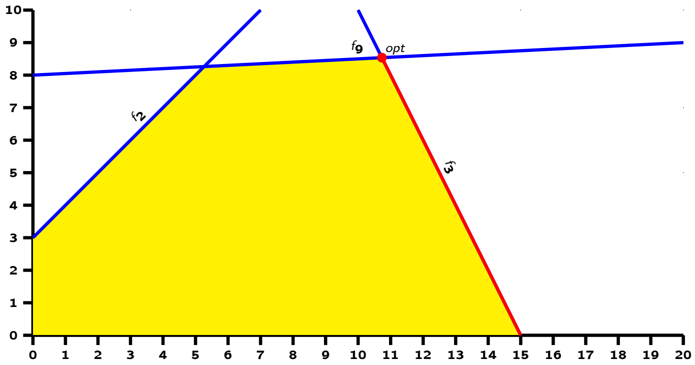
$f_3 : y \leqslant -2 \cdot x + 30$

$f_9 : y \leqslant 1.05 \cdot x + 8$

1. **Falls** $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. **Ansonsten wähle** $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. **Falls** $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. **Ansonsten berechne den Schnitt mit der Hyperebene** h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Beispiel (Aufstieg der letzten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



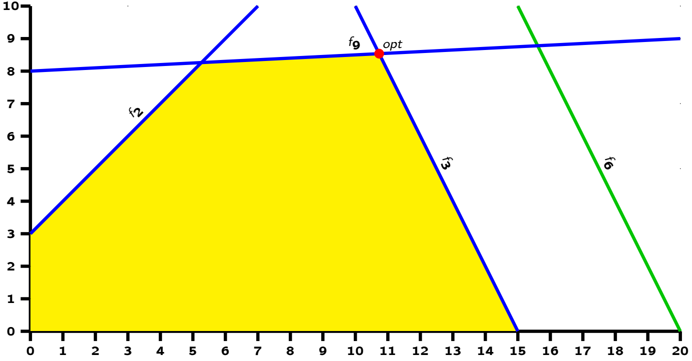
Füge nacheinander wieder ein: f_2, f_3,

$f_2 : y \leqslant 1 \cdot x + 2$
$f_3 : y \leqslant -2 \cdot x + 30$

$f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der letzten vier Rekursionen)



Füge nacheinander wieder ein: f_2, f_3, f_6

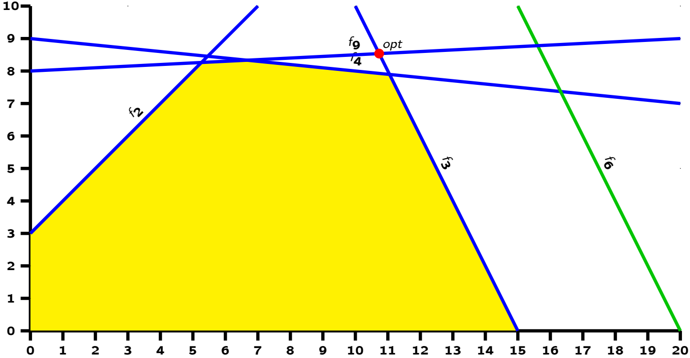
$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3.Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der letzten vier Rekursionen)

d **Variablen,** *m* **Nebenbedingungen, Dimentsion** *n*



Füge nacheinander wieder ein: f_2, f_3, f_6 und f_4.

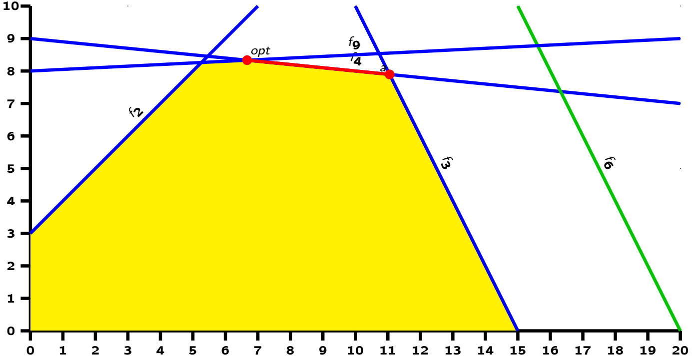
$$f_4 : y \leqslant -0.9 \cdot x + 9$$

$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \qquad f_6 : y \leqslant -2 \cdot x + 40 \qquad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der letzten vier Rekursionen)



Füge nacheinander wieder ein: f_2, f_3, f_6 und f_4.

$$f_4 : y \leqslant -0.9 \cdot x + 9$$

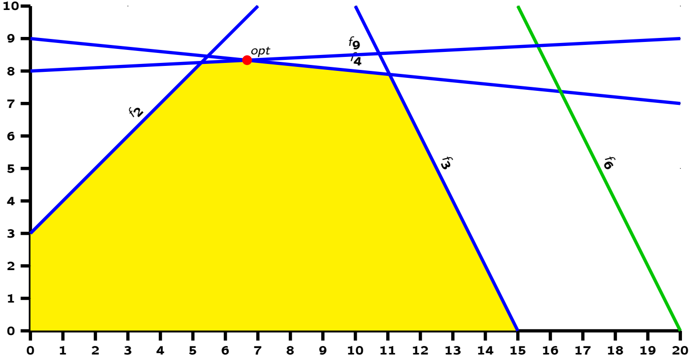
$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der letzten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Füge nacheinander wieder ein: f_2, f_3, f_6 und f_4.

$$f_4 : y \leqslant -0.9 \cdot x + 9$$

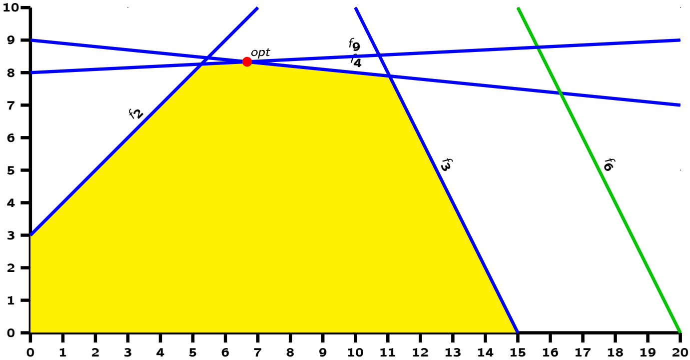
$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der ersten vier Rekursionen)

d **Variablen,** m **Nebenbedingungen, Dimentsion** n



Füge nacheinander wieder ein:

$$f_4 : y \leqslant -0.9 \cdot x + 9$$

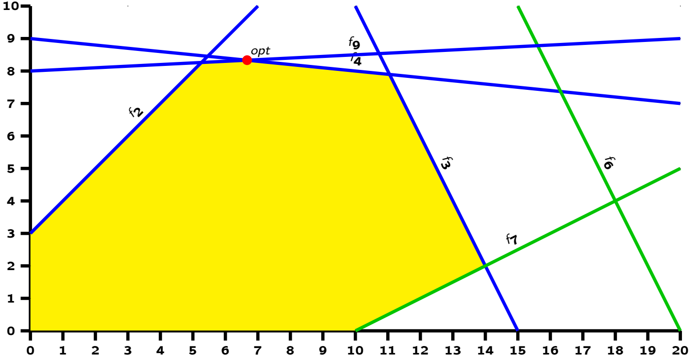
$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

1. **Falls** $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. **Ansonsten wähle** $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. **Falls** $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. **Ansonsten berechne den Schnitt mit der Hyperebene** h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Beispiel (Aufstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Füge nacheinander wieder ein: f_7,

$f_4 : y \leqslant -0.9 \cdot x + 9$ $\quad f_7 : y \geqslant 0.5 \cdot x - 5$

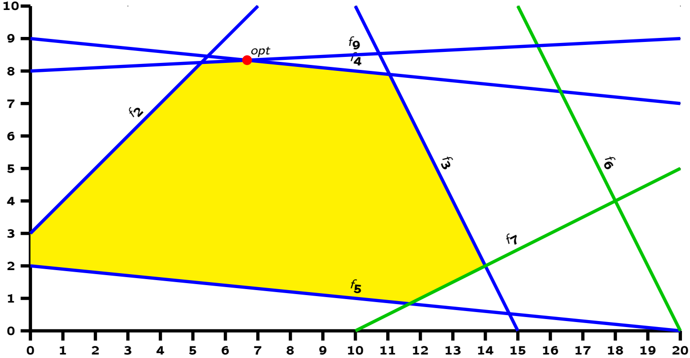
$f_2 : y \leqslant 1 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30$ $\quad f_6 : y \leqslant -2 \cdot x + 40$ $\quad f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Füge nacheinander wieder ein: f_7, f_5,

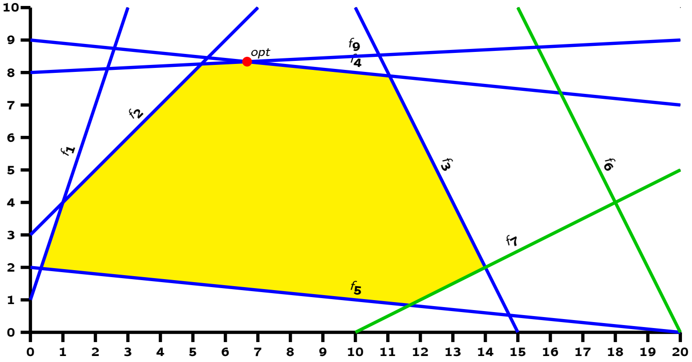
$f_4 : y \leqslant -0.9 \cdot x + 9$ $\quad f_7 : y \geqslant 0.5 \cdot x - 5$

$f_2 : y \leqslant 1 \cdot x + 2$ $\quad f_5 : y \geqslant -0.9 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30$ $\quad f_6 : y \leqslant -2 \cdot x + 40$ $\quad f_9 : y \leqslant 1.05 \cdot x + 8$

Beispiel (Aufstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Füge nacheinander wieder ein: f_7, f_5, f_1

$f_1 : y \leqslant 3 \cdot x + 1$ $f_4 : y \leqslant -0.9 \cdot x + 9$ $f_7 : y \geqslant 0.5 \cdot x - 5$

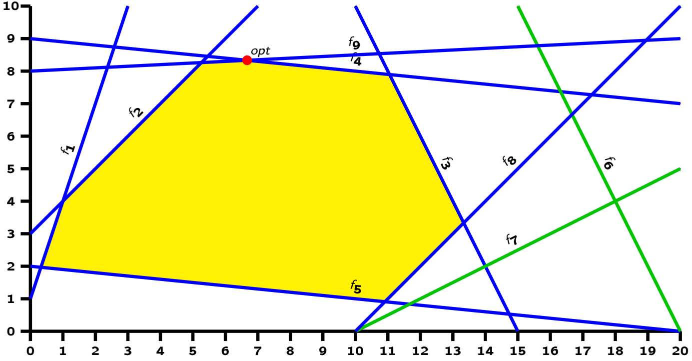
$f_2 : y \leqslant 1 \cdot x + 2$ $f_5 : y \geqslant -0.9 \cdot x + 2$

$f_3 : y \leqslant -2 \cdot x + 30$ $f_6 : y \leqslant -2 \cdot x + 40$ $f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Aufstieg der ersten vier Rekursionen)

d Variablen, m Nebenbedingungen, Dimentsion n



Füge nacheinander wieder ein: f_7, f_5, f_1 und f_8.

$f_1 : y \leqslant 3 \cdot x + 1$	$f_4 : y \leqslant -0.9 \cdot x + 9$	$f_7 : y \geqslant 0.5 \cdot x - 5$
$f_2 : y \leqslant 1 \cdot x + 2$	$f_5 : y \geqslant -0.9 \cdot x + 2$	$f_8 : y \geqslant 1 \cdot x - 10$
$f_3 : y \leqslant -2 \cdot x + 30$	$f_6 : y \leqslant -2 \cdot x + 40$	$f_9 : y \leqslant 1.05 \cdot x + 8$

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.

Laufzeiten und Details (Schritt 1 bis 3)

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.
  - Dies kann optimal mittels Greedy gelöst werden.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.
  - Dies kann optimal mittels Greedy gelöst werden.
  - Laufzeit: $O(d \log d) = O(d^2)$.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.
  - Dies kann optimal mittels Greedy gelöst werden.
  - Laufzeit: $O(d \log d) = O(d^2)$.
- In Schritt 2 wird rekursiv ein Problem gelöst, bei dem m um eins verringert ist.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.
  - Dies kann optimal mittels Greedy gelöst werden.
  - Laufzeit: $O(d \log d) = O(d^2)$.
- In Schritt 2 wird rekursiv ein Problem gelöst, bei dem m um eins verringert ist.
- In Schritt 3 wird getestet, ob die gewählte Hyperebene h nicht die Lösung $opt(H \setminus \{h\})$ verletzt.

Laufzeiten und Details (Schritt 1 bis 3)

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.
  - Dies kann optimal mittels Greedy gelöst werden.
  - Laufzeit: $O(d \log d) = O(d^2)$.
- In Schritt 2 wird rekursiv ein Problem gelöst, bei dem m um eins verringert ist.
- In Schritt 3 wird getestet, ob die gewählte Hyperebene h nicht die Lösung $opt(H \setminus \{h\})$ verletzt.
  - Das kann in $O(d)$ gelöst werden.

Laufzeiten und Details (Schritt 1 bis 3)

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.
  - Dies kann optimal mittels Greedy gelöst werden.
  - Laufzeit: $O(d \log d) = O(d^2)$.
- In Schritt 2 wird rekursiv ein Problem gelöst, bei dem m um eins verringert ist.
- In Schritt 3 wird getestet, ob die gewählte Hyperebene h nicht die Lösung $opt(H \setminus \{h\})$ verletzt.
  - Das kann in $O(d)$ gelöst werden.

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:
    - $h_k^- \leqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ und

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:
    - $h_k^- \leqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ und
    - $h_k^+ \geqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:
    - $h_k^- \leqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ und
    - $h_k^+ \geqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
  - Nenne diese beiden letzten Bedingungen h' und h''.

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:
    - $h_k^- \leqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ und
    - $h_k^+ \geqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
  - Nenne diese beiden letzten Bedingungen h' und h''.
- Setze nun $H' = H \cup \{h', h''\} \setminus \{h\}$ und löse H' rekursiv.

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:
    - $h_k^- \leqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ und
    - $h_k^+ \geqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
  - Nenne diese beiden letzten Bedingungen h' und h''.
- Setze nun $H' = H \cup \{h', h''\} \setminus \{h\}$ und löse H' rekursiv.
- Beachte: H' hat $m + 1$ Nebenbedingungen, aber nur $d - 1$ Variablen.

Laufzeiten und Details (Schritt 4)

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:
    - $h_k^- \leqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ und
    - $h_k^+ \geqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
  - Nenne diese beiden letzten Bedingungen h' und h''.
- Setze nun $H' = H \cup \{h', h''\} \setminus \{h\}$ und löse H' rekursiv.
- Beachte: H' hat $m + 1$ Nebenbedingungen, aber nur $d - 1$ Variablen.

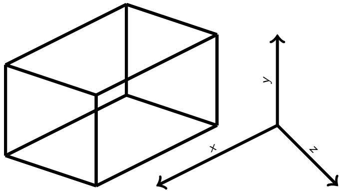
Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10$

$0 \leqslant y, \quad y \leqslant 10$

$0 \leqslant z, \quad z \leqslant 10$



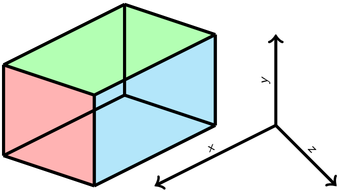
Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10$

$0 \leqslant y, \quad y \leqslant 10$

$0 \leqslant z, \quad z \leqslant 10$



- Gegeben seien die sechs einfachen Nebenbedingungen

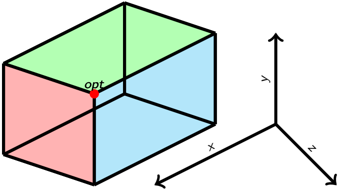
Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10$

$0 \leqslant y, \quad y \leqslant 10$

$0 \leqslant z, \quad z \leqslant 10$



- Gegeben seien die sechs einfachen Nebenbedingungen
- Zu maximieren sei die Funktion $f(x, y, z) = 0.99 \cdot x + 0.974 \cdot y + z$.

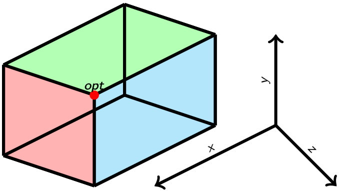
Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10 \quad x + y + z \leqslant 25$

$0 \leqslant y, \quad y \leqslant 10$

$0 \leqslant z, \quad z \leqslant 10$



- Gegeben seien die sechs einfachen Nebenbedingungen
- Zu maximieren sei die Funktion $f(x, y, z) = 0.99 \cdot x + 0.974 \cdot y + z$.
- Das Maximum wird bei $f(10, 10, 10)$ erreicht.

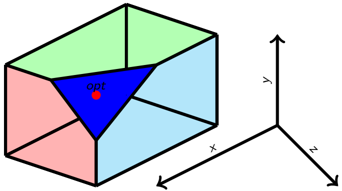
Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10 \quad x + y + z \leqslant 25$

$0 \leqslant y, \quad y \leqslant 10$

$0 \leqslant z, \quad z \leqslant 10$



- Gegeben seien die sechs einfachen Nebenbedingungen
- Zu maximieren sei die Funktion $f(x, y, z) = 0.99 \cdot x + 0.974 \cdot y + z$.
- Das Maximum wird bei $f(10, 10, 10)$ erreicht.
- Die zusätzliche Nebenbedingung $x + y + z \leqslant 25$ widerspricht diesem Maximum

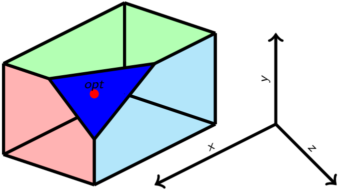
Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10 \quad x + y + z \leqslant 25$

$0 \leqslant y, \quad y \leqslant 10$

$0 \leqslant z, \quad z \leqslant 10$

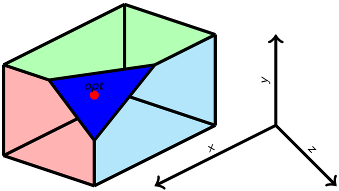


- Gegeben seien die sechs einfachen Nebenbedingungen
- Zu maximieren sei die Funktion $f(x, y, z) = 0.99 \cdot x + 0.974 \cdot y + z$.
- Das Maximum wird bei $f(10, 10, 10)$ erreicht.
- Die zusätzliche Nebenbedingung $x + y + z \leqslant 25$ widerspricht diesem Maximum
- Das neue Maximum liegt damit auf der blauen Fläche.

Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10 \quad x + y + z \leqslant 25$
$0 \leqslant y, \quad y \leqslant 10$
$0 \leqslant z, \quad z \leqslant 10$



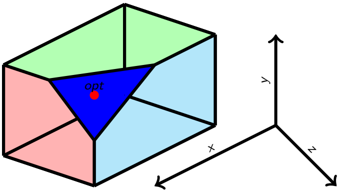
- Gegeben seien die sechs einfachen Nebenbedingungen
- Zu maximieren sei die Funktion $f(x, y, z) = 0.99 \cdot x + 0.974 \cdot y + z$.
- Das Maximum wird bei $f(10, 10, 10)$ erreicht.
- Die zusätzliche Nebenbedingung $x + y + z \leqslant 25$ widerspricht diesem Maximum
- Das neue Maximum liegt damit auf der blauen Fläche.
- Wir lösen nach z auf: $z = 25 - x - y$. Damit verbleibt:

$$0 \leqslant x, \quad x \leqslant 10, \quad x + y \leqslant 25 \quad f(x, y) = -0.01 \cdot x - 0.026 \cdot y + 25$$
$$0 \leqslant y, \quad y \leqslant 10, \quad 15 \leqslant x + y$$

Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$0 \leqslant x, \quad x \leqslant 10 \quad x + y + z \leqslant 25$
$0 \leqslant y, \quad y \leqslant 10$
$0 \leqslant z, \quad z \leqslant 10$



- Gegeben seien die sechs einfachen Nebenbedingungen
- Zu maximieren sei die Funktion $f(x, y, z) = 0.99 \cdot x + 0.974 \cdot y + z$.
- Das Maximum wird bei $f(10, 10, 10)$ erreicht.
- Die zusätzliche Nebenbedingung $x + y + z \leqslant 25$ widerspricht diesem Maximum
- Das neue Maximum liegt damit auf der blauen Fläche.
- Wir lösen nach z auf: $z = 25 - x - y$. Damit verbleibt:

$$0 \leqslant x, \quad x \leqslant 10, \quad x + y \leqslant 25 \quad f(x, y) = -0.01 \cdot x - 0.026 \cdot y + 25$$
$$0 \leqslant y, \quad y \leqslant 10, \quad 15 \leqslant x + y$$

Korrektheit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

1. **Falls** $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. **Ansonsten wähle** $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. **Falls** $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. **Ansonsten berechne den Schnitt mit der Hyperebene** h**, und löse das so entstandene** $(d - 1)$**-dimensionale LP rekursiv.**

Korrektheit

- Setze $k(d, m) = 2 \cdot d + m$.

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Korrektheit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Korrektheit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Korrektheit

d **Variablen**, m **Nebenbedingungen, Dimentsion** n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d - 1)$-dimensionale LP rekursiv.

Korrektheit

d Variablen, m Nebenbedingungen, Dimentsion n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.
- Falls h eine Nebenbedingung ist, die keinen Einfluss auf die optimale Lösung hat.

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3.Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Korrektheit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.
- Falls h eine Nebenbedingung ist, die keinen Einfluss auf die optimale Lösung hat.
  - Dann bestimmt der Algorithmus $opt(H \setminus \{h\})$ in Schritt 3.

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3.Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Korrektheit

d Variablen, m Nebenbedingungen, Dimentsion n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.
- Falls h eine Nebenbedingung ist, die keinen Einfluss auf die optimale Lösung hat.
  - Dann bestimmt der Algorithmus $opt(H \setminus \{h\})$ in Schritt 3.
- Falls h eine Nebenbedingung ist, mit der sich die optimale Lösung schneidet.

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Korrektheit

d **Variablen,** m **Nebenbedingungen, Dimentsion** n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.
- Falls h eine Nebenbedingung ist, die keinen Einfluss auf die optimale Lösung hat.
  - Dann bestimmt der Algorithmus $opt(H \setminus \{h\})$ in Schritt 3.
- Falls h eine Nebenbedingung ist, mit der sich die optimale Lösung schneidet.
  - Dann befindet sich die optimale Lösung auf der Hyperebene, die durch h beschrieben wird.

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3.Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Korrektheit

d **Variablen,** m **Nebenbedingungen, Dimentsion** n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.
- Falls h eine Nebenbedingung ist, die keinen Einfluss auf die optimale Lösung hat.
  - Dann bestimmt der Algorithmus $opt(H \setminus \{h\})$ in Schritt 3.
- Falls h eine Nebenbedingung ist, mit der sich die optimale Lösung schneidet.
  - Dann befindet sich die optimale Lösung auf der Hyperebene, die durch h beschrieben wird.
  - Diese betrachtet der Algorithmus in Schritt 4.

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d - 1)$-dimensionale LP rekursiv.

Korrektheit

d **Variablen,** m **Nebenbedingungen, Dimentsion** n

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.
- Falls h eine Nebenbedingung ist, die keinen Einfluss auf die optimale Lösung hat.
  - Dann bestimmt der Algorithmus $opt(H \setminus \{h\})$ in Schritt 3.
- Falls h eine Nebenbedingung ist, mit der sich die optimale Lösung schneidet.
  - Dann befindet sich die optimale Lösung auf der Hyperebene, die durch h beschrieben wird.
  - Diese betrachtet der Algorithmus in Schritt 4.

1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
2. Ansonsten wähle $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
4. Ansonsten berechne den Schnitt mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Laufzeit

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.

Laufzeit

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:

Laufzeit

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.

Laufzeit

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.
  - Schritt 2: $T_2 = T(m-1, d)$.

Laufzeit

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.
  - Schritt 2: $T_2 = T(m-1, d)$.
  - Schritt 3: $T_3 = O(d)$.

Laufzeit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Sei $T(m,d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.
  - Schritt 2: $T_2 = T(m-1,d)$.
  - Schritt 3: $T_3 = O(d)$.
  - Schritt 4: $T_4 = T(m+1,d-1) + O(d \cdot m)$.

Laufzeit

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.
  - Schritt 2: $T_2 = T(m-1, d)$.
  - Schritt 3: $T_3 = O(d)$.
  - Schritt 4: $T_4 = T(m+1, d-1) + O(d \cdot m)$.
- Dabei wird der Schritt 4 nicht immer ausgeführt.

Laufzeit

- Sei $T(m,d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.
  - Schritt 2: $T_2 = T(m-1,d)$.
  - Schritt 3: $T_3 = O(d)$.
  - Schritt 4: $T_4 = T(m+1,d-1) + O(d \cdot m)$.
- Dabei wird der Schritt 4 nicht immer ausgeführt.
- Im Folgenden schätzen wir diese Wahrscheinlichkeit ab.

Laufzeit

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.
  - Schritt 2: $T_2 = T(m-1, d)$.
  - Schritt 3: $T_3 = O(d)$.
  - Schritt 4: $T_4 = T(m+1, d-1) + O(d \cdot m)$.
- Dabei wird der Schritt 4 nicht immer ausgeführt.
- Im Folgenden schätzen wir diese Wahrscheinlichkeit ab.

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.
- Sei D eine Auswahl mit $|D| = d$ dieser x^* bestimmenden Hyperebenen.

Aussage

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.
- Sei D eine Auswahl mit $|D| = d$ dieser x^* bestimmenden Hyperebenen.
- Damit gilt: $opt(D) = x^*$.

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.
- Sei D eine Auswahl mit $|D| = d$ dieser x^* bestimmenden Hyperebenen.
- Damit gilt: $opt(D) = x^*$.
- Schritt 4 wird ausgeführt, falls $opt(H \setminus \{h\})$ die Nebenbedingung h verletzt.

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.
- Sei D eine Auswahl mit $|D| = d$ dieser x^* bestimmenden Hyperebenen.
- Damit gilt: $opt(D) = x^*$.
- Schritt 4 wird ausgeführt, falls $opt(H \setminus \{h\})$ die Nebenbedingung h verletzt.
- D.h. Schritt 4 wird ausgeführt, falls $h \in D$.

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.
- Sei D eine Auswahl mit $|D| = d$ dieser x^* bestimmenden Hyperebenen.
- Damit gilt: $opt(D) = x^*$.
- Schritt 4 wird ausgeführt, falls $opt(H \setminus \{h\})$ die Nebenbedingung h verletzt.
- D.h. Schritt 4 wird ausgeführt, falls $h \in D$.
- Damit ist die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird:

$$\mathbb{P}\mathrm{r}[h \in D] = \frac{|D \cap H|}{|H|} \leqslant \frac{d}{m}.$$

Aussage

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.
- Sei D eine Auswahl mit $|D| = d$ dieser x^* bestimmenden Hyperebenen.
- Damit gilt: $opt(D) = x^*$.
- Schritt 4 wird ausgeführt, falls $opt(H \setminus \{h\})$ die Nebenbedingung h verletzt.
- D.h. Schritt 4 wird ausgeführt, falls $h \in D$.
- Damit ist die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird:

$$\mathbb{P}\mathrm{r}[h \in D] = \frac{|D \cap H|}{|H|} \leqslant \frac{d}{m}.$$

Abschätzung der Laufzeit

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m, d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.

Abschätzung der Laufzeit

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m, d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:

Abschätzung der Laufzeit

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m,d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:
- Im Falle $m > 0$ und $d > 1$ gilt:

$$T(m,d) \leqslant T(m-1,d) + d^2 + \frac{d}{m} \cdot T(m+1,d-1).$$

Abschätzung der Laufzeit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m,d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:
- Im Falle $m > 0$ und $d > 1$ gilt:

$$T(m,d) \leqslant T(m-1,d) + d^2 + \frac{d}{m} \cdot T(m+1,d-1).$$

- Im Falle $m = 0$ oder $d = 1$ gilt:

$$T(m,d) \leqslant d^2 + m.$$

Abschätzung der Laufzeit

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m,d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:
- Im Falle $m > 0$ und $d > 1$ gilt:

$$T(m,d) \leqslant T(m-1,d) + d^2 + \frac{d}{m} \cdot T(m+1,d-1).$$

- Im Falle $m = 0$ oder $d = 1$ gilt:

$$T(m,d) \leqslant d^2 + m.$$

- Wir definieren nun $f(1) = 1$ und für $d > 1$:

$$f(d) = d \cdot f(d-1) + 3 \cdot d^3$$

Abschätzung der Laufzeit

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m,d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:
- Im Falle $m > 0$ und $d > 1$ gilt:

$$T(m,d) \leqslant T(m-1,d) + d^2 + \frac{d}{m} \cdot T(m+1,d-1).$$

- Im Falle $m = 0$ oder $d = 1$ gilt:

$$T(m,d) \leqslant d^2 + m.$$

- Wir definieren nun $f(1) = 1$ und für $d > 1$:

$$f(d) = d \cdot f(d-1) + 3 \cdot d^3$$

- Damit gilt:

$$f(d) = d! + \sum_{k=2}^{d} 3 \cdot k^3 \cdot \frac{d!}{(k-1)!} = O(d!).$$

Abschätzung der Laufzeit

d **Variablen**, *m* **Nebenbedingungen**, **Dimentsion** *n*

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m,d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:
- Im Falle $m > 0$ und $d > 1$ gilt:

$$T(m,d) \leqslant T(m-1,d) + d^2 + \frac{d}{m} \cdot T(m+1,d-1).$$

- Im Falle $m = 0$ oder $d = 1$ gilt:

$$T(m,d) \leqslant d^2 + m.$$

- Wir definieren nun $f(1) = 1$ und für $d > 1$:

$$f(d) = d \cdot f(d-1) + 3 \cdot d^3$$

- Damit gilt:

$$f(d) = d! + \sum_{k=2}^{d} 3 \cdot k^3 \cdot \frac{d!}{(k-1)!} = O(d!).$$

- Denn $\sum_{k=2}^{d} \frac{3 \cdot k^3}{(k-1)!}$ ist durch Konstante beschränkt.

Abschätzung der Laufzeit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m, d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:
- Im Falle $m > 0$ und $d > 1$ gilt:

$$T(m, d) \leqslant T(m-1, d) + d^2 + \frac{d}{m} \cdot T(m+1, d-1).$$

- Im Falle $m = 0$ oder $d = 1$ gilt:

$$T(m, d) \leqslant d^2 + m.$$

- Wir definieren nun $f(1) = 1$ und für $d > 1$:

$$f(d) = d \cdot f(d-1) + 3 \cdot d^3$$

- Damit gilt:

$$f(d) = d! + \sum_{k=2}^{d} 3 \cdot k^3 \cdot \frac{d!}{(k-1)!} = O(d!).$$

- Denn $\sum_{k=2}^{d} \frac{3 \cdot k^3}{(k-1)!}$ ist durch Konstante beschränkt.

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1) \cdot f(d) + 2 \cdot d^2$.

Beweis:

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m, d) \leqslant (m-1) \cdot f(d) + 2 \cdot d^2$.

Beweis:

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m, d) \leqslant (m-1) \cdot f(d) + 2 \cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m, d) \leqslant d^2 + 1 \leqslant (m-1) \cdot f(d) + 2 \cdot d^2.$$

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1) \cdot f(d) + 2 \cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m,d) \leqslant d^2 + 1 \leqslant (m-1) \cdot f(d) + 2 \cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m,d) \leqslant m + 1 \leqslant (m-1) \cdot f(d) + 2 \cdot d^2.$$

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1)\cdot f(d) + 2\cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m,d) \leqslant d^2 + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m,d) \leqslant m + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Für den Induktionsschritt setzen wir:

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1)\cdot f(d) + 2\cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m,d) \leqslant d^2 + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m,d) \leqslant m + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Für den Induktionsschritt setzen wir:
  - $k = 2\cdot d + m$

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1)\cdot f(d) + 2\cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m,d) \leqslant d^2 + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m,d) \leqslant m + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Für den Induktionsschritt setzen wir:
  - $k = 2\cdot d + m$
  - Es sei nun $m \geqslant 2$ und $d \geqslant 2$.

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1)\cdot f(d) + 2\cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m,d) \leqslant d^2 + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m,d) \leqslant m + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Für den Induktionsschritt setzen wir:
  - $k = 2\cdot d + m$
  - Es sei nun $m \geqslant 2$ und $d \geqslant 2$.
  - Behauptung sei gezeigt für alle d' und m' mit $2\cdot d' + m' < k$.

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1)\cdot f(d) + 2\cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m,d) \leqslant d^2 + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m,d) \leqslant m + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Für den Induktionsschritt setzen wir:
  - $k = 2\cdot d + m$
  - Es sei nun $m \geqslant 2$ und $d \geqslant 2$.
  - Behauptung sei gezeigt für alle d' und m' mit $2\cdot d' + m' < k$.
  - Also auch für $(m-1, d)$ und $(m+1, d-1)$.

Finale Abschätzung der Laufzeit

Lemma

Es gilt: $T(m,d) \leqslant (m-1)\cdot f(d) + 2\cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m,d) \leqslant d^2 + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m,d) \leqslant m + 1 \leqslant (m-1)\cdot f(d) + 2\cdot d^2.$$

- Für den Induktionsschritt setzen wir:
  - $k = 2\cdot d + m$
  - Es sei nun $m \geqslant 2$ und $d \geqslant 2$.
  - Behauptung sei gezeigt für alle d' und m' mit $2\cdot d' + m' < k$.
  - Also auch für $(m-1, d)$ und $(m+1, d-1)$.

Induktionsschluss

$T(m, d) \leqslant (m - \mathbf{1}) \cdot f(d) + \mathbf{2} \cdot d^{\mathbf{2}}$
$f(d) = d \cdot f(d - \mathbf{1}) + \mathbf{3} \cdot d^{\mathbf{3}}$

$$T(m, d) \leqslant T(m-1, d) + d^2 + \frac{d}{m} \cdot T(m+1, d-1)$$

Induktionsschluss

$T(m, d) \leqslant (m - \mathbf{1}) \cdot f(d) + \mathbf{2} \cdot d^{\mathbf{2}}$
$f(d) = d \cdot f(d - \mathbf{1}) + \mathbf{3} \cdot d^{\mathbf{3}}$

$$
\begin{aligned}
T(m,d) &\leqslant T(m-1,d) + d^2 + \tfrac{d}{m} \cdot T(m+1, d-1) \\
&\leqslant (m-2) \cdot f(d) + 2 \cdot d^2 + d^2 + \tfrac{d}{m} \cdot T(m+1, d-1)
\end{aligned}
$$

Induktionsschluss

$T(m,d) \leqslant (m-\mathbf{1}) \cdot f(d) + \mathbf{2} \cdot d^{\mathbf{2}}$
$f(d) = d \cdot f(d-\mathbf{1}) + \mathbf{3} \cdot d^{\mathbf{3}}$

$$\begin{aligned} T(m,d) &\leqslant T(m-1,d) + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\ &\leqslant (m-2) \cdot f(d) + 2 \cdot d^2 + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\ &\leqslant (m-2) \cdot f(d) + 3 \cdot d^2 + \tfrac{d}{m} \cdot m \cdot f(d-1) + 2 \cdot (d-1)^2 \end{aligned}$$

Induktionsschluss

$T(m,d) \leqslant (m-\mathbf{1}) \cdot f(d) + \mathbf{2} \cdot d^{\mathbf{2}}$
$f(d) = d \cdot f(d-\mathbf{1}) + \mathbf{3} \cdot d^{\mathbf{3}}$

$$
\begin{aligned}
T(m,d) &\leqslant T(m-1,d) + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\
&\leqslant (m-2) \cdot f(d) + 2 \cdot d^2 + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\
&\leqslant (m-2) \cdot f(d) + 3 \cdot d^2 + \tfrac{d}{m} \cdot m \cdot f(d-1) + 2 \cdot (d-1)^2 \\
&\leqslant (m-2) \cdot f(d) + 3 \cdot d^2 + \tfrac{d}{m} \cdot m \cdot f(d-1) + 2 \cdot d^2
\end{aligned}
$$

Induktionsschluss

$T(m,d) \leqslant (m-\mathbf{1}) \cdot f(d) + \mathbf{2} \cdot d^{\mathbf{2}}$
$f(d) = d \cdot f(d-\mathbf{1}) + \mathbf{3} \cdot d^{\mathbf{3}}$

$$
\begin{aligned}
T(m,d) &\leqslant T(m-1,d) + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\
&\leqslant (m-2) \cdot f(d) + 2 \cdot d^2 + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\
&\leqslant (m-2) \cdot f(d) + 3 \cdot d^2 + \tfrac{d}{m} \cdot m \cdot f(d-1) + 2 \cdot (d-1)^2 \\
&\leqslant (m-2) \cdot f(d) + 3 \cdot d^2 + \tfrac{d}{m} \cdot m \cdot f(d-1) + 2 \cdot d^2 \\
&= (m-2) \cdot f(d) + 3 \cdot d^2 + d \cdot \tfrac{f(d)-\mathbf{3}\cdot d^{\mathbf{3}}}{d} + 2 \cdot d^2
\end{aligned}
$$

Induktionsschluss

$T(m,d) \leqslant (m-\mathbf{1}) \cdot f(d) + \mathbf{2} \cdot d^{\mathbf{2}}$
$f(d) = d \cdot f(d-\mathbf{1}) + \mathbf{3} \cdot d^{\mathbf{3}}$

$$
\begin{aligned}
T(m,d) &\leqslant T(m-1,d) + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\
&\leqslant (m-2) \cdot f(d) + 2 \cdot d^2 + d^2 + \tfrac{d}{m} \cdot T(m+1,d-1) \\
&\leqslant (m-2) \cdot f(d) + 3 \cdot d^2 + \tfrac{d}{m} \cdot m \cdot f(d-1) + 2 \cdot (d-1)^2 \\
&\leqslant (m-2) \cdot f(d) + 3 \cdot d^2 + \tfrac{d}{m} \cdot m \cdot f(d-1) + 2 \cdot d^2 \\
&= (m-2) \cdot f(d) + 3 \cdot d^2 + d \cdot \tfrac{f(d) - 3 \cdot d^3}{d} + 2 \cdot d^2 \\
&= (m-2) \cdot f(d) + 3 \cdot d^2 + f(d) - 3 \cdot d^3 + 2 \cdot d^2
\end{aligned}
$$

Induktionsschluss

$T(m,d) \leqslant (m-1)\cdot f(d) + 2\cdot d^2$
$f(d) = d \cdot f(d-1) + 3\cdot d^3$

$$\begin{aligned}
T(m,d) &\leqslant T(m-1,d) + d^2 + \tfrac{d}{m}\cdot T(m+1,d-1)\\
&\leqslant (m-2)\cdot f(d) + 2\cdot d^2 + d^2 + \tfrac{d}{m}\cdot T(m+1,d-1)\\
&\leqslant (m-2)\cdot f(d) + 3\cdot d^2 + \tfrac{d}{m}\cdot m\cdot f(d-1) + 2\cdot(d-1)^2\\
&\leqslant (m-2)\cdot f(d) + 3\cdot d^2 + \tfrac{d}{m}\cdot m\cdot f(d-1) + 2\cdot d^2\\
&= (m-2)\cdot f(d) + 3\cdot d^2 + d\cdot \tfrac{f(d)-3\cdot d^3}{d} + 2\cdot d^2\\
&= (m-2)\cdot f(d) + 3\cdot d^2 + f(d) - 3\cdot d^3 + 2\cdot d^2\\
&\leqslant (m-1)\cdot f(d) + 2\cdot d^2
\end{aligned}$$

Aussage

Theorem

Der Algorithmus von Seidel löst ein zulässiges d-dimensionales LP mit m Nebenbedingungen in erwarteter Laufzeit von $O(m \cdot d!)$.

Aussage

Theorem

Der Algorithmus von Seidel löst ein zulässiges d-dimensionales LP mit m Nebenbedingungen in erwarteter Laufzeit von $O(m \cdot d!)$.

- Ist d konstant, so ist die erwartete Laufzeit $O(m)$.

Aussage

Theorem

Der Algorithmus von Seidel löst ein zulässiges d-dimensionales LP mit m Nebenbedingungen in erwarteter Laufzeit von $O(m \cdot d!)$.

- Ist d konstant, so ist die erwartete Laufzeit $O(m)$.
- Ist $d > 10$ so ist die Konstante aber ein wenig unpraktisch.

Aussage

Theorem

Der Algorithmus von Seidel löst ein zulässiges d-dimensionales LP mit m Nebenbedingungen in erwarteter Laufzeit von $O(m \cdot d!)$.

- Ist d konstant, so ist die erwartete Laufzeit $O(m)$.
- Ist $d > 10$ so ist die Konstante aber ein wenig unpraktisch.

Literatur

- B. Korte, J. Vygen. Combinatorial Optimization: Theory and Algorithms, 2nd Edition, Springer, 2002.

Literatur

- B. Korte, J. Vygen. Combinatorial Optimization: Theory and Algorithms, 2nd Edition, Springer, 2002.
- E. Lawler. Combinatorial Optimization: Networks and Matroids. Dover Publications, 1976.

Literatur

- B. Korte, J. Vygen. Combinatorial Optimization: Theory and Algorithms, 2nd Edition, Springer, 2002.
- E. Lawler. Combinatorial Optimization: Networks and Matroids. Dover Publications, 1976.
- C. Papadimitriou und K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice Hall, 1982.

Literatur

- B. Korte, J. Vygen. Combinatorial Optimization: Theory and Algorithms, 2nd Edition, Springer, 2002.
- E. Lawler. Combinatorial Optimization: Networks and Matroids. Dover Publications, 1976.
- C. Papadimitriou und K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice Hall, 1982.
- A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003.

Literatur

- B. Korte, J. Vygen. Combinatorial Optimization: Theory and Algorithms, 2nd Edition, Springer, 2002.
- E. Lawler. Combinatorial Optimization: Networks and Matroids. Dover Publications, 1976.
- C. Papadimitriou und K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice Hall, 1982.
- A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003.