Lineare Programmierung

Übersicht

1. Problemformulierung und Beispiel
2. Inkrementeller, determ. Algorithmus
3. Randomisierter Algorithmus
4. Unbeschränkte lineare Programme
5. Höhere Dimensionen

Lineare Programmierung

Problem:

Maximiere $c_1x_1 + c_2x_2 + \cdots + c_dx_d$

unter den Nebenbedingungen:

$$
\begin{aligned}
a_{1,1}x_1 + \cdots + a_{1,d}x_d &\leq b_1 \\
a_{2,1}x_1 + \cdots + a_{2,d}x_d &\leq b_2 \\
&\vdots \\
a_{n,1}x_1 + \cdots + a_{n,d}x_d &\leq b_n
\end{aligned}
$$

Lineares Programm der Dimension d

$\vec{c} = (c_1, c_2, \ldots, c_d)$

$h_i = \left\{(x_1, \ldots, x_d) \mid a_{i,1}x_1 + \cdots + a_{i,d}x_d \leq b_i\right\}$

$\ell_i =$ Hyperebene, die h_i begrenzt
(Gerade, falls $d = 2$)

$H = \{h_1, \ldots, h_n\}$

Lineare Programmierung

Beispiel

Produktion von zwei Gütern A und B unter Verwendung von vier Rohstoffen

Preis für A: 6 WE   Preis für B: 3 WE

	Rst. 1	Rst. 2	Rst. 3	Rst. 4
Prod. A	2	2	6	2
Prod. B	4	1	2	2
Reserv.	5	2	4	3

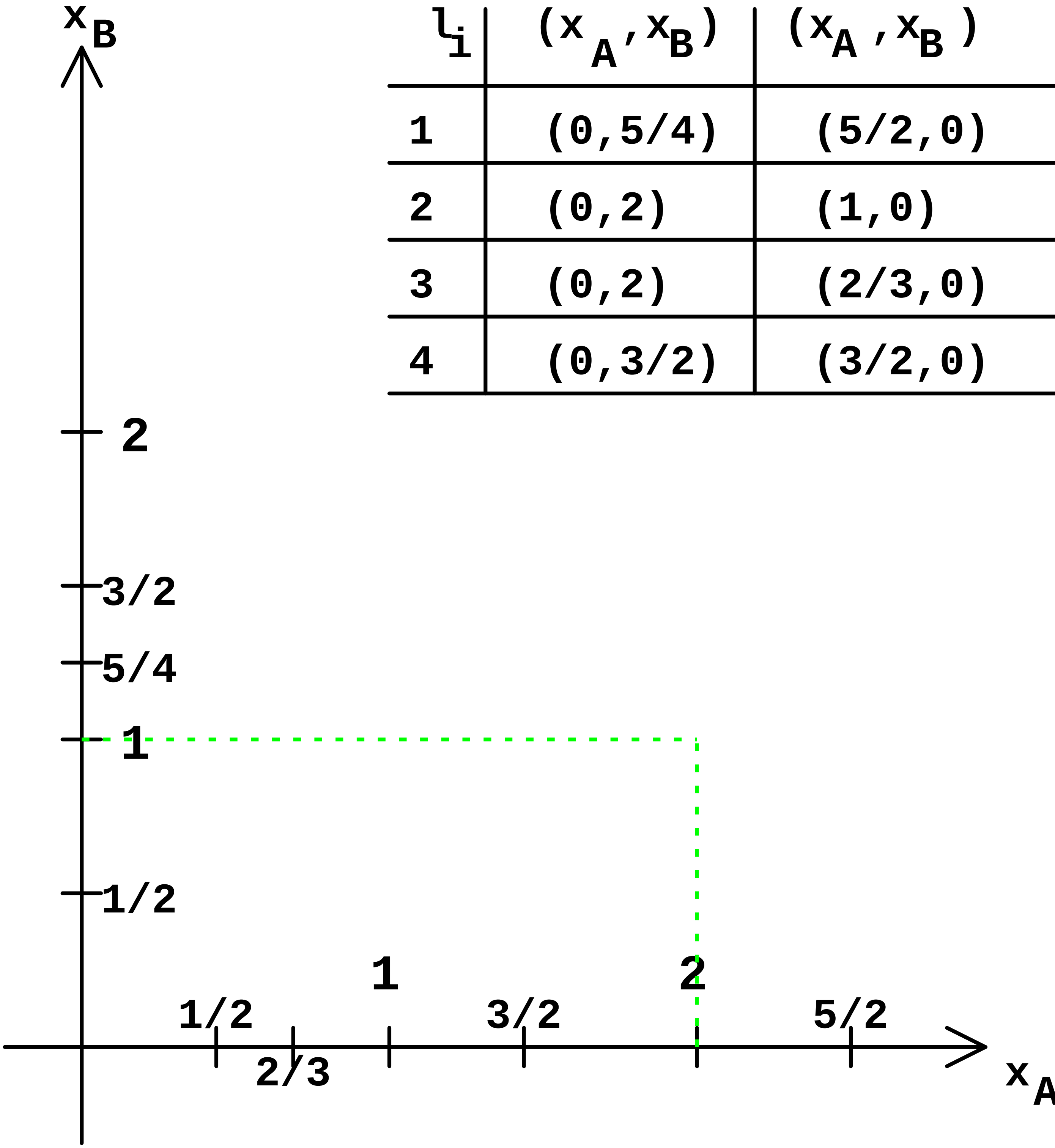
Maximiere Gewinn $\quad 6x_A + 3x_B$

unter den Nebenbedingungen:

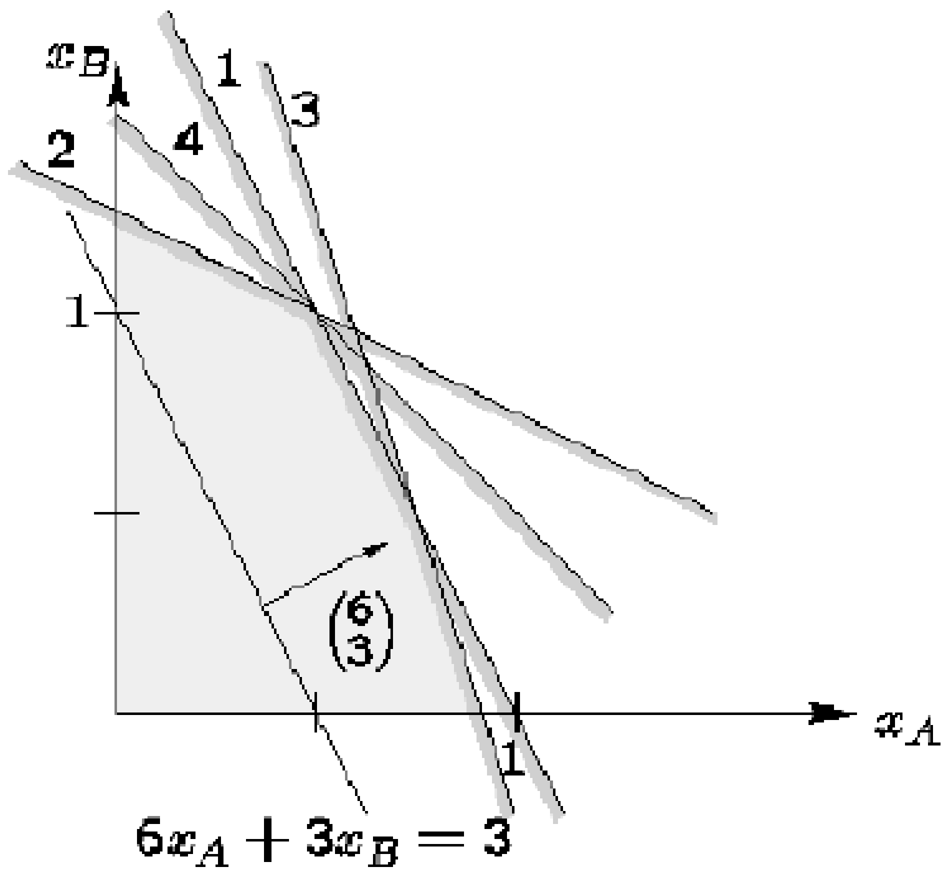
$$\begin{aligned} 2x_A + \ 4x_B &\leq 5 \\ 2x_A + \ \ x_B &\leq 2 \\ 6x_A + \ 2x_B &\leq 4 \\ 2x_A + \ 2x_B &\leq 3 \\ x_A, \ \ x_B &\geq 0 \end{aligned}$$

	$x_A=0, x_B$	$x_A, x_B=0$

l_i	(x_A, x_B)	(x_A, x_B)
1	(0,5/4)	(5/2,0)
2	(0,2)	(1,0)
3	(0,2)	(2/3,0)
4	(0,3/2)	(3/2,0)

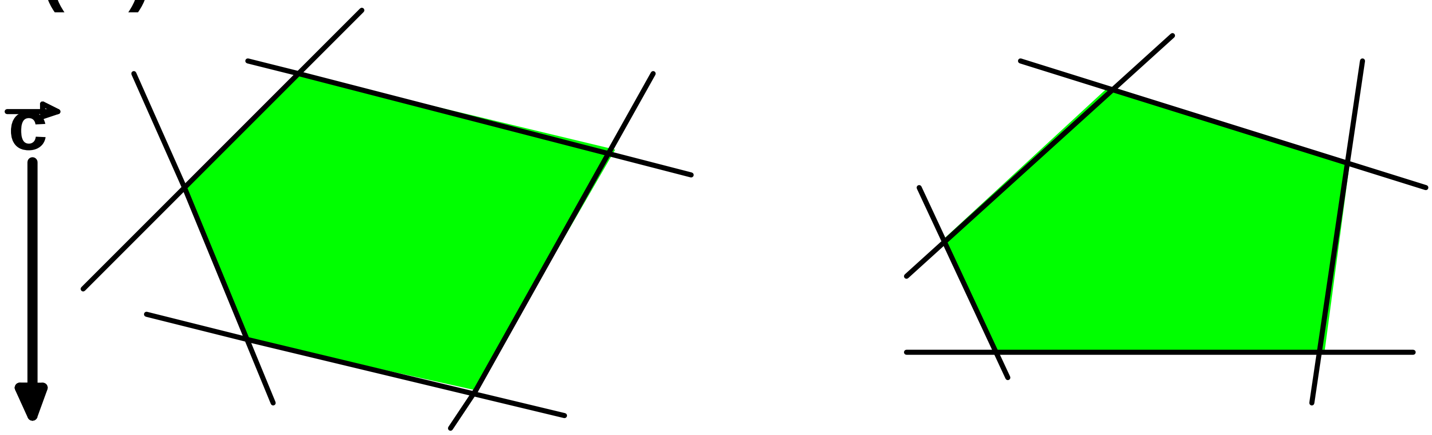


Lineare Programmierung
Beispiel

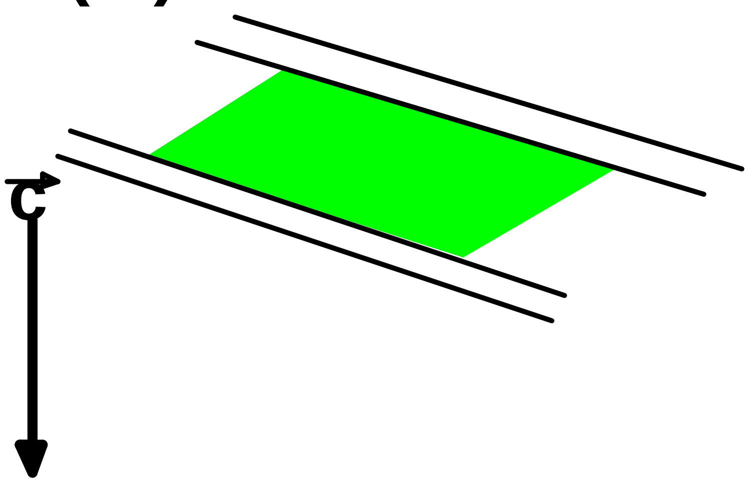


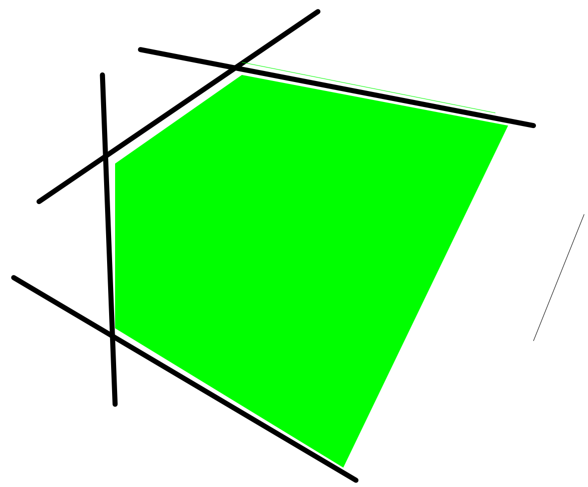
Struktur des zulässigen Bereichs

(1) beschränkt

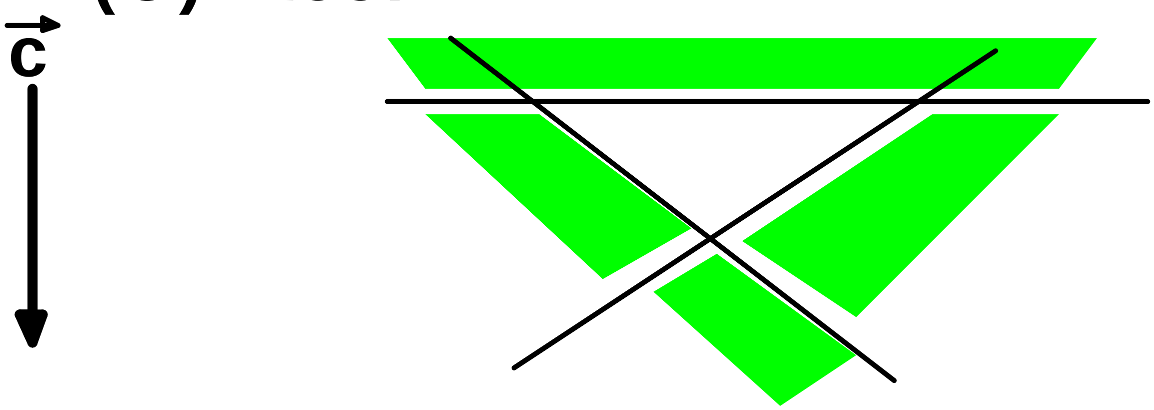


(2) unbeschränkt





(3) leer



Lineare Programmierung

Vier Möglichkeiten für die Lösung eines linearen Programms

1. Eine Ecke des zulässigen Bereichs ist die einzige Lösung.
2. Eine Kante des zulässigen Bereichs enthält alle Lösungen.
3. Es gibt keine zulässige Lösung.
4. Der zulässige Bereich ist unbeschränkt in Richtung $\vec{c}$.

2. Fall: lexikographisch minimale Lösung $\Rightarrow$ Ecke

Lineare Programmierung
Beschränkte Progamme

Annahme:
Algorithmus *LPunbeschränkt*$(H, \vec{c})$ liefert entweder

a) Einen Strahl in $\bigcap H$, der in Richtung $\vec{c}$ unbeschränkt ist, oder

b) Zwei Halbebenen h_1 und h_2, so daß $h_1 \cap h_2$ in Richtung $\vec{c}$ beschränkt ist, oder

c) die Antwort, daß $LP(H, \vec{c})$ keine Lösung hat, weil der zulässige Bereich leer ist

Lineare Programmierung

Sei

$$C_2 = h_1 \cap h_2$$

Restl. Halbebenen: $h_3, \ldots, h_n$

$$C_i = C_{i-1} \cap h_i = h_1 \cap \cdots \cap h_i$$

Berechne-opt-Ecke$(H, \vec{c})$

$v_2 := \ell_1 \cap \ell_2$
$C_2 := h_1 \cap h_2$
for $i := 3$ **to** n **do**
    $C_i := C_{i-1} \cap h_i$
    $v_i :=$ optimale Ecke von C_i

$$C_2 \supseteq C_3 \supseteq C_4 \cdots \supseteq C_n = C$$

$$C_i = \emptyset \quad \Rightarrow \quad C = \emptyset$$

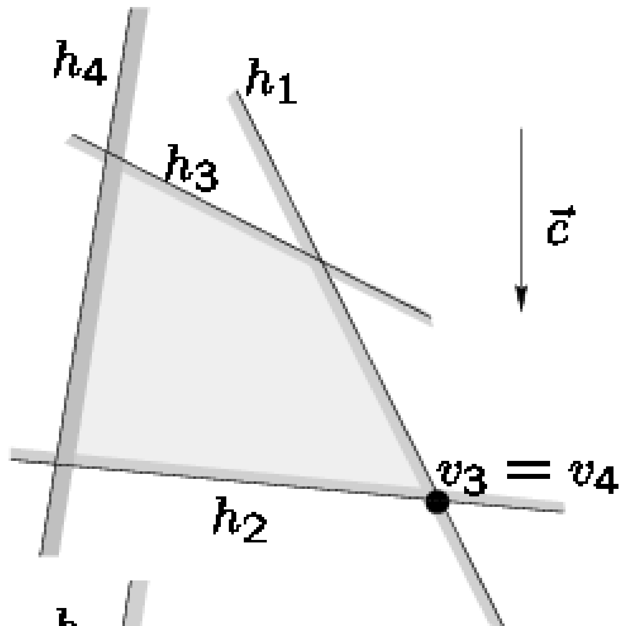
Lineare Programmierung
Optimale Ecke

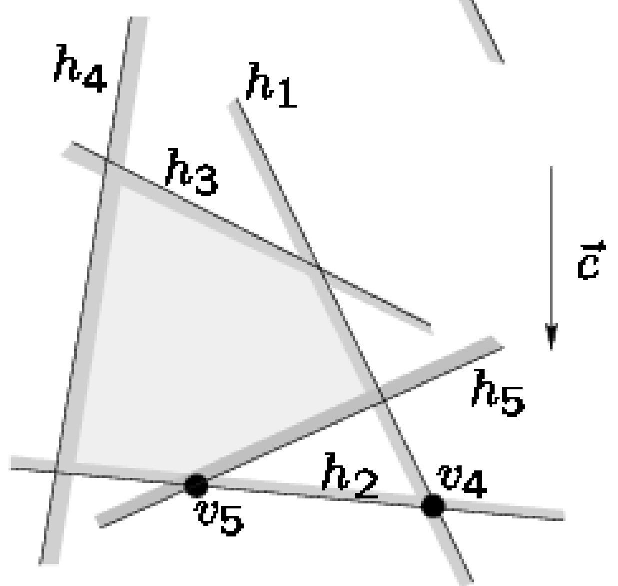
Lemma 1 Sei $3 \leq i \leq n$. Dann gilt:

1. Wenn $v_{i-1} \in h_i$ ist, dann ist $v_i = v_{i-1}$.

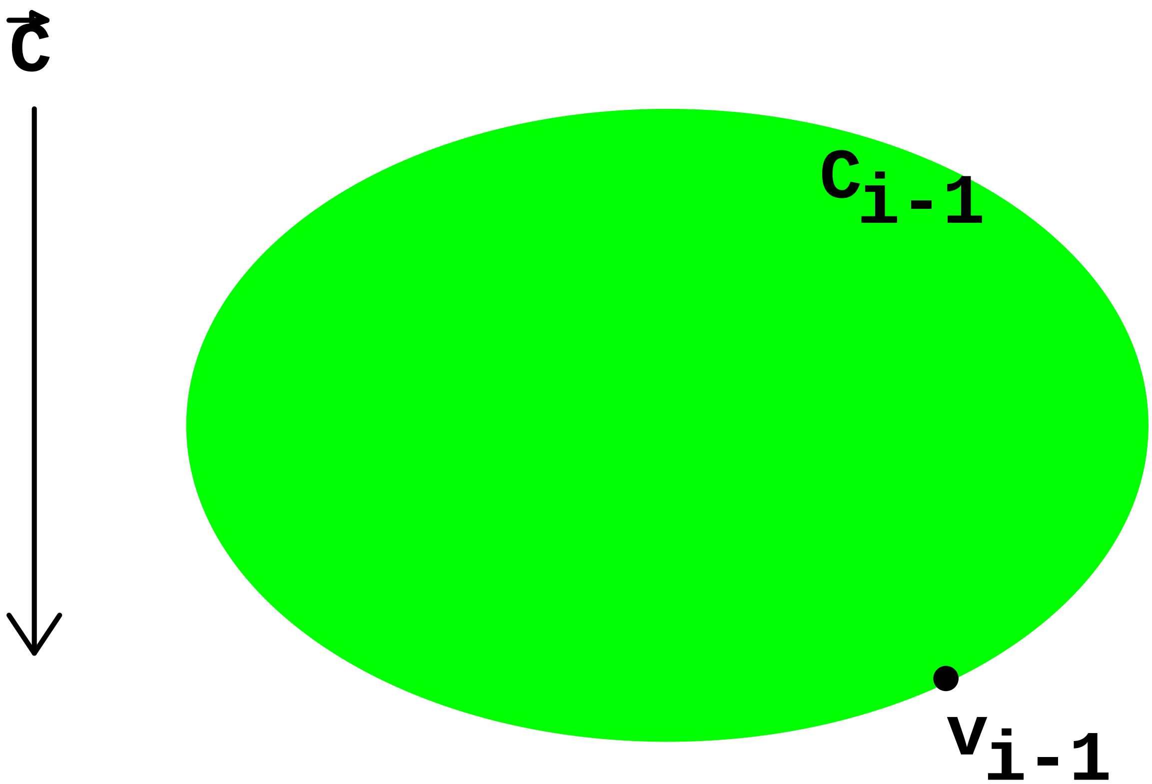
2. Wenn $v_{i-1} \notin h_i$ ist, dann ist entweder $C_i = \emptyset$ oder v_i liegt auf ℓ_i.

Lineare Programmierung
Optimale Ecke





$f_c(x)=c_1x_1+c_2x_2$



Lineare Programmierung
Optimale Ecke

Neues Problem:

Finde den Punkt auf l_i, der $\vec{c}x$ unter den Bedingungen $x \in h_j$, $1 \leq j \leq i-1$, maximiert.

Beobachtung: $l_i \cap h_j$ ist ein Strahl

Sei $S_{i-1} := \{h_1 \cap l_i, \ldots, h_{i-1} \cap l_i\}$

1-dim-LP$(S_{i-1}, \vec{c})$

```
ρ_1 = s_1
for j := 2 to i − 1 do
    ρ_j := ρ_{j−1} ∩ s_j
if ρ_{i−1} ≠ ∅
    then return die optimale Ecke von ρ_{i−1}
    else return ∅
```

$\Rightarrow$ Zeit $O(i)$

Lineare Programmierung

Programm

Algorithmus *2D-LP*
Input: Ein 2-dim. lineares Programm $(H, \vec{c})$
Output: Entweder eine optimale Ecke oder $\emptyset$ oder ein Strahl in $\bigcap H$, der in Richtung $\vec{c}$ unbeschränkt ist

```
if LPunbeschränkt(H, c⃗) ≠ {h, h'}
   then return LPunbeschränkt(H, c⃗)
h₁ := h, h₂ := h'; v₂ := ℓ₁ ∩ ℓ₂
h₃, ..., hₙ := restl. Halbebenen in H
for i := 3 to n do
    if v_{i-1} ∈ h_i
       then v_i := v_{i-1}
       else  S_{i-1} := H_{i-1} ∩* ℓ_i
             v_i := 1-dim-LP(S_{i-1}, c⃗)
    if v_i existiert nicht
       then return ∅
return v_n
```

Laufzeit $O(n^2)$

Lineare Programmierung
Reihenfolge