(Ganzzahlige) Lineare Programme

Vorlesung Theoretische Informatik

Sommersemester 2003

Party im Kaufleuten...

Vorhanden sind:

- 2000 Einheiten Champagner
- 5000 Einheiten Bier
- 6000 Einheiten Mineralwasser

Potentielle Gäste:

- VIP-Gäste
- Szene-Gäste

Gesucht: Anweisungen an Türsteher, wieviele Gäste welcher Kategorie einzulassen sind, damit

- die Getränke ausreichen
- der Spasswert maximiert wird

Getränkekonsum und Spasswert

- **Erwarteter Konsum in Einheiten:**

	VIP	Szene
Champagner	3	1
Bier	3	4
Mineralwasser	2	5

- **Spasswert:** ein VIP-Gast ist doppelt so viel wert wie ein Szene-Gast. Spassfunktion gegeben durch

$$f(v, s) = 2v + s,$$

wobei

  - v Anzahl der VIP-Gäste
  - s Anzahl der Szene-Gäste

Ein bedingtes Optimierungsproblem

Unbekannte: v, s.

Ziel: maximiere Spassfunktion $f(v, s) = 2v + s$ unter den Nebenbedingungen

$$\begin{array}{rcrcrl}
3v & + & s & \leq & 2000 & \text{(genug Champagner)} \\
3v & + & 4s & \leq & 5000 & \text{(genug Bier)} \\
2v & + & 5s & \leq & 6000 & \text{(genug Mineralwasser)} \\
v & & & \geq & 0 & \text{(Gästezahl nicht negativ...)} \\
 & & s & \geq & 0 & \text{(Gästezahl nicht negativ...)} \\
v & & & \in & \mathbb{Z} & \text{(... und ganzzahlig)} \\
 & & s & \in & \mathbb{Z} & \text{(... und ganzzahlig)}
\end{array}$$

Einige Gästeaufteilungen:

v	s	$f(v, s)$	Interpretation
0	1200	1200	(nur Szene)
666	0	1332	(nur VIPs)

Wir werden sehen: die Mischung macht's!

Eine (unzulässige?) Vereinfachung

Ignoriere die Ganzzahligkeitsbedingungen!

Maximiere $f(v,s) = 2v + s$ unter

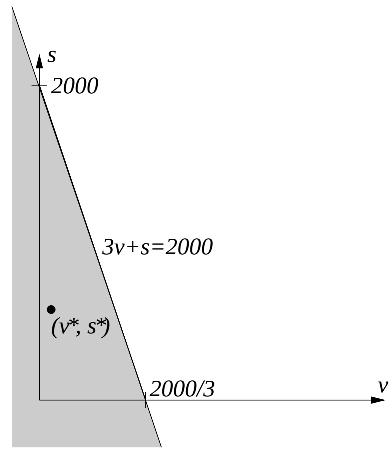
$$\begin{array}{rcrcr} 3v & + & s & \leq & 2000 \\ 3v & + & 4s & \leq & 5000 \\ 2v & + & 5s & \leq & 6000 \\ v & & & \geq & 0 \\ & & s & \geq & 0 \end{array}$$

Werden sehen:

- dieses 'relaxierte' Problem ist einfach zu lösen, aber
- aus der relaxierten Lösung eine ganzzahlige Optimallösung zu rekonstruieren, ist im allgemeinen schwierig!

Lösungsansatz: Graphische Methode

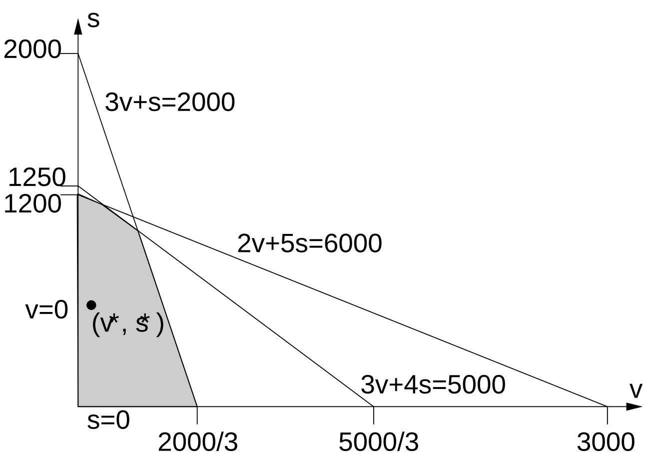
Beobachtung: Ein Paar (v^*, s^*) erfüllt genau dann die erste Nebenbedingung, wenn (v^*, s^*) ein Punkt der Ebene unterhalb oder auf der Geraden mit Gleichung $3v + s = 2000$ ist.
($\Rightarrow$ **Halbebene**)



Analog für die anderen Nebenbedingungen!
($\Rightarrow$ **5 Halbebenen**)

Der Zulässigkeitsbereich

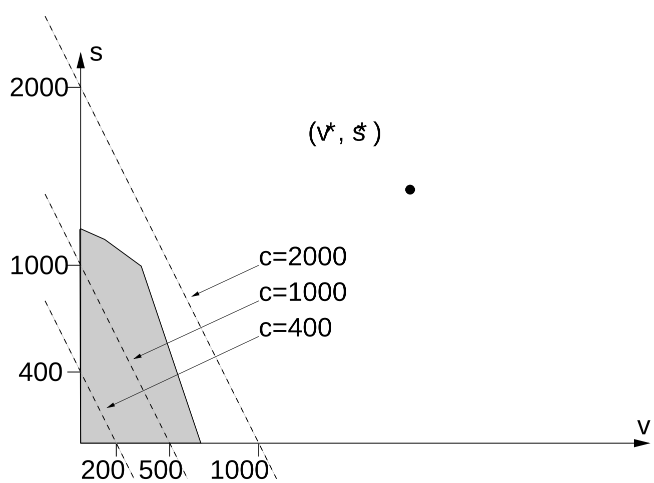
Definition: Die Menge aller (v^*, s^*), die alle Nebenbedingungen erfüllen, heisst *Zulässigkeitsbereich*.



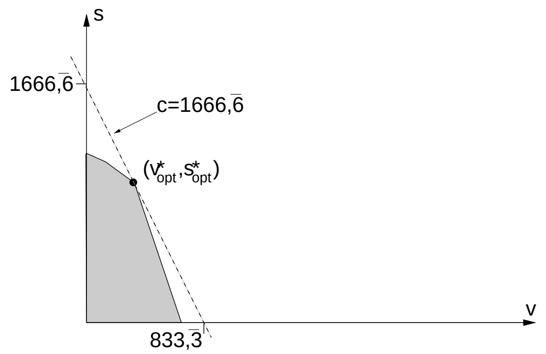
Zulässigkeitsbereich ist ein *Polygon* (kann im allgemeinen leer oder unbeschränkt sein).

Wo liegt das Optimum?

Beobachtung: Für jedes c haben die Punkte (v^*, s^*) auf der Geraden mit Gleichung $2v+s = c$ den gleichen Spasswert $f(v^*, s^*) = c$.



$\Rightarrow$ finde grösstes c, so dass die Gerade $2v+s = c$ noch den Zulässigkeitsbereich schneidet!



(v^*_{opt}, s^*_{opt}) ist Schnitt der beiden Geraden

$$\begin{array}{rcrcl} 3v & + & 4s & = & 5000 \quad \text{und} \\ 3v & + & s & = & 2000 \\ \hline & & 3s & = & 3000 \\ & \Rightarrow & s^*_{opt} & = & 1000 \\ & & v^*_{opt} & = & \frac{2000-s}{3} = 333,\overline{3} \end{array}$$

Getränkekonsum:

	VIPs	Szene	Total
Champagner	$3 \cdot 333,\overline{3}$	$1 \cdot 1000$	2000
Bier	$3 \cdot 333,\overline{3}$	$4 \cdot 1000$	5000
Mineralwasser	$2 \cdot 333,\overline{3}$	$5 \cdot 1000$	$5666,\overline{6}$

$$\begin{array}{rcl} s^*_{opt} & = & 1000 \\ v^*_{opt} & = & 333,\overline{3} \\ f(v^*, s^*) & = & 1666,\overline{6} \end{array}$$

Rundungsversuche:

v	s	$f(v,s)$	
333	1000	1666	
334	999	1667	Champagner reicht nicht
334	998	1666	

Hier: Glück gehabt: optimaler Spasswert bei ganzzahliger Lösung ist höchstens

$$\lfloor f(v^*, s^*) \rfloor = \lfloor 1666,\overline{6} \rfloor = 1666,$$

und dieser Wert ist erreichbar.

Im allgemeinen: Runden auf eine ganzzahlige Optimallösung ist schwierig!

Lineare Programme (I)

Kaufleuten-Problem (ohne Ganzzahligkeitsbedingungen) ist ein *lineares Programm*.

Allgemein:

- n Unbekannte

$$x_1, \ldots, x_n$$

- *Lineare* Zielfunktion

$$c_1x_1 + c_2x_2 + \cdots c_nx_n$$

- m *lineare* Nebenbedingungen

$$\begin{array}{ccccccccc} a_{1,1}x_1 & + & a_{1,2}x_2 & + & \cdots & + & a_{1,n}x_n & \leq & b_1 \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + & \cdots & + & a_{2,n}x_n & \leq & b_2 \\ & & & & \vdots & & & & \\ a_{m,1}x_1 & + & a_{m,2}x_2 & + & \cdots & + & a_{m,n}x_n & \leq & b_m \end{array}$$

Die $c_i, a_{j,i}$ und b_j sind Konstanten.

Lineare Programme (II)

Ziel: maximiere $c_1x_1 + c_2x_2 + \cdots c_nx_n$ unter den Nebenbedingungen

$$\begin{array}{ccccccccc} a_{1,1}x_1 & + & a_{1,2}x_2 & + & \cdots & + & a_{1,n}x_n & \leq & b_1 \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + & \cdots & + & a_{2,n}x_n & \leq & b_2 \\ & & & & \vdots & & & & \\ a_{m,1}x_1 & + & a_{m,2}x_2 & + & \cdots & + & a_{m,n}x_n & \leq & b_m \end{array}$$

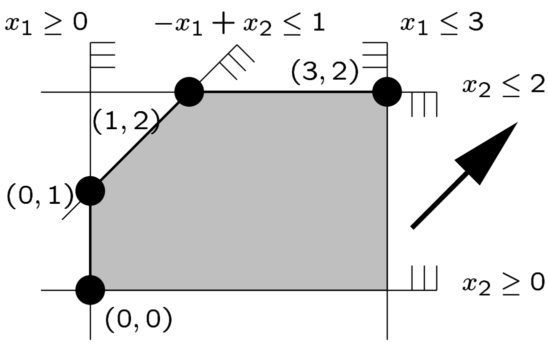
Definition:

- Die Menge aller n-Tupel $(x^*_1, x^*_2, \ldots, x^*_n)$, die alle Nebenbedingungen erfüllen (zulässige Lösungen), heisst *Zulässigkeitsbereich*
  - kann leer sein (Problem *unlösbar*)
  - kann unbeschränkt sein

- Eine zulässige Lösung $(x^*_1, x^*_2, \ldots, x^*_n)$ mit maximalem Wert $c_1x^*_1 + c_2x^*_2 + \cdots c_nx^*_n$ heisst *optimale Lösung*
  - kann im unbeschränkten Fall existieren, muss aber nicht

Weitere Beispiele (I)

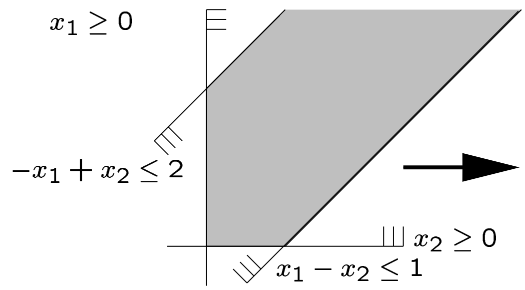
$$
\begin{array}{llrcrcl}
\text{maximiere} & x_1 + x_2 & & & & & \\
\text{unter} & & -x_1 & + & x_2 & \leq & 1, \\
& & x_1 & & & \leq & 3, \\
& & & & x_2 & \leq & 2, \\
& x_1, x_2 \geq 0. & & & & &
\end{array}
$$



Ecke $(3, 2)$ des Zulässigkeitsbereichs ist die eindeutige optimale Lösung.

Weitere Beispiele (II)

$$
\begin{array}{llrcrcl}
\text{maximiere} & & x_1 & & & & \\
\text{unter} & & x_1 & - & x_2 & \leq & 1, \\
& & -x_1 & + & x_2 & \leq & 2, \\
& x_1, x_2 \geq 0. & & & & &
\end{array}
$$



Zulässigkeitsbereich unbeschränkt, und es existiert keine optimale Lösung.

Weitere Beispiele (III)

Nicht immer nur zwei Unbekannte:

$$
\begin{array}{lllllllll}
\text{maximiere} & x_0 & & & & & & & \\
\text{unter} & x_1 & + & x_2 & + \cdots + & x_n & \leq & 1 - x_0 \\
& x_1 & & & & & \geq & x_0 \\
& & & x_2 & & & \geq & x_0 \\
& & & & & & \vdots & \\
& & & & & x_n & \geq & x_0
\end{array}
$$

. . . und:

- Eine Ungleichung kann beliebig "sortiert" sein und auf "$\leq$" oder "$\geq$" lauten
- Minimieren ist auch erlaubt!
- Als Nebenbedingungen dürfen auch Gleichungen auftreten

Übung: Gib eine optimale Lösung für obiges Problem an! Was ist der optimale Zielfunktionswert?

Lösen eines Linearen Programms (I)

Graphische Methode scheidet bei grösseren Problemen (und mehr als zwei Unbekannten) aus.

Theorem: (Khachiyan 1979) Ein lineares Programm mit n Variablen, m Ungleichungen und U der maximalen Bitgrösse einer Eingabezahl kann in Zeit poly(n, m, U) gelöst werden.

$\Rightarrow$ Lineares Programmieren ist in $\mathbf{P}$.

Beweis: komplizierter Algorithmus (sogenannte *Ellipsoidmethode*), nicht ohne grossen Aufwand implementierbar.

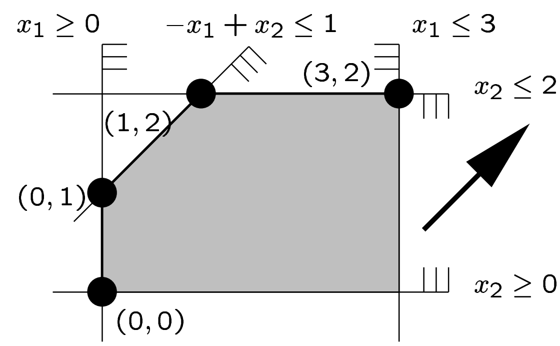
Lösen eines Linearen Programms (II)

Der sogenannte *Simplex-Algorithmus* ist ein sehr einfacher Algorithmus zum Lösen linearer Programme,

- der in der Praxis sehr schnell ist,
- dessen Laufzeit aber in der Theorie noch nicht analysiert werden kann,
- von dem insbesondere nicht bekannt ist, ob er polynomielle Laufzeit hat.

Idee des Simplex-Algorithmus

Optimale Lösung ist eine Ecke des Zulässigkeitsbereichs.



- Beginne bei irgendeiner Ecke des Zulässigkeitsbereichs
- gehe solange zu einer benachbarten Ecke mit höherem Zielfunktionswert, bis die optimale Ecke erreicht wird.

Ganzzahlige Lineare Programme

Kaufleuten-Problem (mit Ganzzahligkeitsbedingungen) ist ein *ganzzahliges lineares Programm*.

- Lineare Programme (LP) sind in Polynomialzeit lösbar

- Ganzzahlige lineare Programme (ILP) sind NP-schwer

Relaxierung: ILP → LP durch Weglassen der Ganzzahligkeitsbedingungen

Runden: Berechnung einer (im allgemeinen nicht optimalen) Lösung für das ILP aus einer optimalen Lösung für das LP

Lineare Programme in der Praxis

- Lineares Programmieren ist oft Vorverarbeitungsschritt für die (approximative) Lösung NP-schwerer Probleme (die oft als ILP formuliert werden können)

- Es gibt kommerzielle Löser für lineare Programme (z.B. CPLEX),die meistens auf dem Simplex-Algorithmus basieren.

- Lineares Programmieren verbraucht einen signifikanten Anteil der weltweit verwendeten Rechenzeit!