Lineare Programmierung

Inhalt

- Rückblick
- Motivation - linearen Programmierung
  - Flussprobleme
  - Multiple Warenflüsse
- Fortsetzung Simplex Algorithmus
  - Initialisierung
  - Fundamentalsatz der linearen Programmierung
  - schwache Dualität
  - Dualität der linearen Programmierung

Norman Dziengel
John Wiesel

Lineare Programmierung

Rückblick (1)

- Vorbereitung Simplex-Algorithmus

Gesucht ist Lösung, die folgende lin. Funktion in Standardform:

$$\sum_{j=1}^{n} c_j x_j$$

unter den Nebenbedingungen

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i \text{ und } i=1,2,...,m$$

$$x_j \geqslant 0 \text{ für } j=1,2,...,n$$

maximiert.

Beispiel:
maximiere $40x_1+50x_2$

Nebenbedingung:

(1) $2x_1 + 3x_2 <= 3$
(2) $4x_1 + 2x_2 <= {}^5/_2$

(3) $x_1, x_2 >= 0$

Lineare Programmierung

Rückblick (2)

- Vorbereitung Simplex-Algorithmus

Standardform:
maximiere $40x_1 + 50x_2$

Nebenbedingung:

(1) $2x_1 + 3x_2 <= 3$

(2) $4x_1 + 2x_2 <= {}^5/_2$

(3) $x_1, x_2 >= 0$

$$s = b_i - \sum_{j=1}^{n} a_{ij} x_j$$

nach Schlupfform

Schlupfform:
$z = 40x_1 + 50x_2$

Nebenbedingung:

(1) $x_3 = 3 - 2x_1 - 3x_2$

(2) $x_4 = {}^5/_2 - 4x_1 - 2x_2$

(3) $x_1, x_2, x_3, x_4 >= 0$

Lineare Programmierung

Rückblick (3)

- Simplex-Algorithmus – eine Iteration

$z = 40x_1 + 50x_2$

Nebenbedingung:

$(1)\ x_3 = 3 - 2x_1 - 3x_2$

$(2)\ x_4 = \frac{5}{2} - 4x_1 - 2x_2$

$(3)\ x_1, x_2, x_3 x_4 \geqslant 0$

Maximiere x_1

$x_1\ maximal\ \frac{3}{2}$

$x_1\ maximal\ \frac{5}{8}$ => eng

=> Basisvariablentausch bei (2) x_4 gegen x_1

$z = 40(\frac{5}{8} - \frac{1}{2}x_2 - \frac{1}{4}x_4) + 50x_2 = 25 + 30x_2 - 10x_4$

$(1)\ x_3 = 3 - 2(\frac{5}{8} - \frac{1}{2}x_2 - \frac{1}{4}x_4) - 3x_2 = \frac{7}{4} - 2x_2 + \frac{1}{2}x_4$

$(2)\ x_1 = \frac{5}{8} - \frac{1}{2}x_2 - \frac{1}{4}x_4$

etc...

Lineare Programmierung

Rückblick (4)

- Simplex-Algorithmus
  - Was wissen wir bereits?

Funktionsweise

Iteration über Gleichungssysteme

- äquivalente Umformungen von einer Schlupfform in die nächste
- Zielfunktionswert steigt in der Regel von Iteration zu Iteration (bei Maximierung)

Eingabe: Schlupfform eines lin. Programms

Ausgabe: optimale Lösung

Laufzeit: meist polynomial

Lineare Programmierung

Rückblick (5)

- Simplex-Algorithmus
  - Was wissen wir bereits?

Terminierung

- kreiselt nach $\binom{n+m}{m}$ Iterationen, weil unbeschränkt
- terminiert mit zulässiger Lösung

n : Anzahl der nicht-Basisvariablen

m : Anzahl der Basisvariablen

Lineare Programmierung

Was bleibt zu zeigen?

- Simplex-Algorithmus
  - Umgang mit LP ohne zulässige Lösung
  - Umgang mit ungültiger initialer Basislösung
  - Liefert „Simplex“ immer optimales Ergebnis?

Lineare Programmierung

Motivation - lineare Programmierung

- Wieso lineare Programmierung, Rechtfertigung?
  - Lösung für viele reale Probleme
    - Flussprobleme,

      z.B. für die Erstellung von Flugplänen, USA 80er Jahre
    - Kürzeste Pfade
    - Kostenkonstellationen in der Betriebswirtschaft
  - Sobald ein Problem einmal als lineares Programm mit polynomialer Größe formuliert ist, kann es mit „Simplex“ gelöst werden.

Lineare Programmierung

Motivation - lineare Programmierung

- Multiple Warenflüsse (1)

(besser bekannt als „multi“ bzw. „multiple commodity flow“)

Die Firma Lucky Puck möchte mehrere Produkte vertreiben. Zu den Pucks kommen nun Schläger, Helme und anderes Zubehör. Dabei müssen täglich die Waren von der entsprechenden Fabrik zum entsprechenden Großhändler transportiert werden.

Als Transportmittel stehen LKWs zur Verfügung, die sich die Autobahnen und Bundestraßen rund um die Fabrik von Lucky Puck teilen müssen.

Lineare Programmierung

Motivation - lineare Programmierung

- Multiple Warenflüsse (2)

Gerichteter Graph G = (V,E)

Jede Kante (u,v) e E besitzt Kapazität c(u,v) >= 0.

Bsp.: k Waren, Ware i mit (s_i, t_i, b_i)
s = source (Quelle), t = target (Senke), d = demand (Bedarf)

$$f(u,v) = \sum_{i=1}^{k} f_i(u,v)$$ (Aggregat-Fluss)

Wie sieht nun lineares Programm dazu aus?

Lineare Programmierung

Motivation - lineare Programmierung

- Multiple Warenflüsse (3)

da keine Optimierung nötig => Zielfunktion: minimiere 0

Nebenbedingungen:

$$\sum_{i=1}^{k} f_i(u,v) \leq c(u,v) \quad \text{für alle } u,v \in V \ ,$$

$$f_i(u,v) = -f_i(v,u) \quad \text{für alle } i = 1,2,\ldots,k \text{ und für alle } u,v \in V \ ,$$

$$\sum_{v \in V} f_i(u,v) = 0 \quad \text{für alle } i = 1,2,\ldots,k \text{ und für alle } u \in V - \{s_i,t_i\} \ ,$$

$$\sum_{v \in V} f_i(s_i,v) = d_i \quad \text{für alle } i = 1,2,\ldots,k \ ,$$

$$f_i(u,v) \leq c(u,v) \quad \text{für alle } u,v \in V \text{ und für alle } i = 1,2,\ldots,k \ .$$

wichtig: „Simplex“ effektivster bekannter Algorithmus

Lineare Programmierung

Simplex-Algorithmus – mögliche Probleme

Bisherige Annahme:

Lineares Programm besitzt eine Lösung.

Lineares Programm besitzt eine initiale Basislösung.

Aber, möglich ist auch:

- initiale Basislösung ungültig,
- LP unlösbar.

Lineare Programmierung

Simplex-Algorithmus – ungültige initiale Basislösung

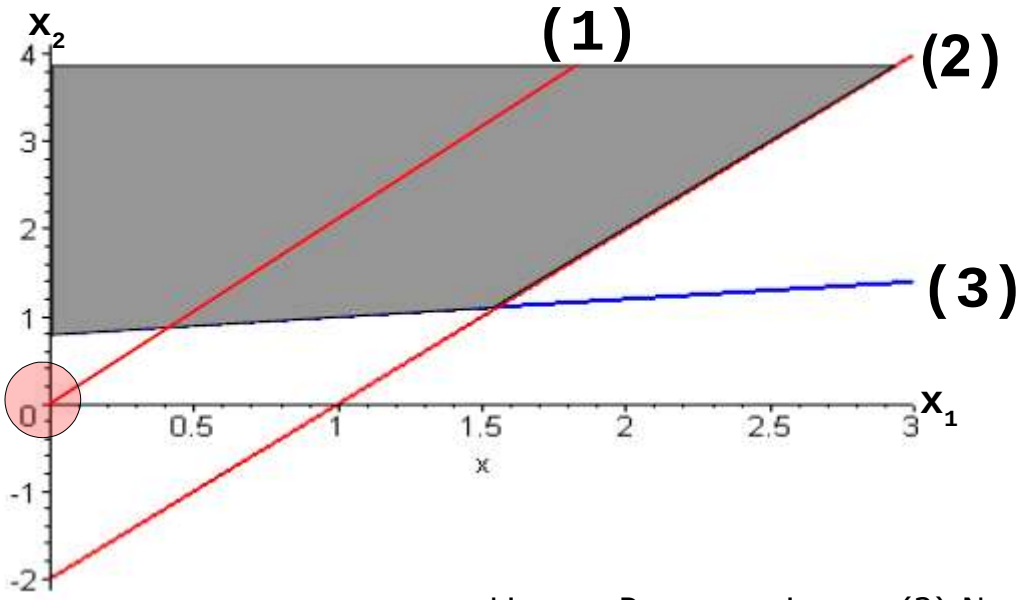
- initiale Basislösung ungültig – Beispiel

(1) maximiere $2x_1 - x_2$

(2) $2x_1 - x_2 <= 2$

(3) $x_1 - 5x_2 <= -4$

$x_1, x_2 >= 0$



(1) $z = 2x_1 - x_2$

(2) $x_3 = 2 - 2x_1 + x_2$

(3) $x_4 = -4 - x_1 + 5x_2$

$x_1, x_2, x_3, x_4 >= 0$

Basislösung: {0,0,-4!,2}

Lineare Programmierung

Simplex-Algorithmus – unlösbar

- Nebenbedingungen im „Widerspruch“
- Unbeschränkt

Vor der Ausführung von „Simplex“ müssen also die Voraussetzungen für einen fehlerfreien Ablauf des Simplex-Algorithmus geschaffen werden.

Lineare Programmierung

Simplex-Algorithmus – Initialisierung

Wir benötigen eine Initialisierungsprozedur für jedes Lineare Programm L, wir nennen es „INITIALIZE-SIMPLEX“ (nach Cormen).

INITIALIZE-SIMPLEX

Schlupfform an Testprozedur übergeben

- Basislösung zulässig: Rückgabe der original Schlupfform
- Basislösung nicht zulässig: weitergeben an Hilfsfunktion

Lineare Programmierung

Simplex-Algorithmus – Initialisierung (2)

„Hilfsfunktion“: ein Lineares Programm, dass prüft, ob L lösbar ist.

L_h hat zusätzlich zu den n Variablen von L, die Hilfsvariable x_0.

maximieren $-x_0$

(b : Basisvariablen, a : Koeffizienten, x : Nicht-Basisvariblen)

$$\sum_{j=1}^{n} a_{ij} x_j - x_0 \leqslant b_i \qquad \text{für } i = 1, 2, ..., m$$

$$x_j \geqslant 0 \qquad \text{für } j = 0, 1, ..., n$$

Wenn der optimale Zielfunktionswert des Hilfsprogramms negativ ist, dann hat das ursprüngliche lineare Programm keine zulässige Lösung. Ist die Lösung gleich 0, dann hat das ursprüngliche LP eine Lösung.

Lineare Programmierung

Simplex-Algorithmus – Initialisierung (3)

INITIALIZE-SIMPLEX

Schlupfform an Testprozedur übergeben

- Basislösung zulässig: Rückgabe der original Schlupfform
- Basislösung nicht zulässig: weitergeben an Hilfsfunktion
  - wenn Basislösung $x_0 = 0$: Rückgabe der original Schlupfform
  - sonst: Rückgabe „L nicht lösbar“

Lineare Programmierung

Fundamentalsatz der linearen Programmierung

Für jedes lineare Programm L, dass in Standardform gegeben ist, gilt genau eine der folgenden drei Aussagen:

1. L besitzt eine optimale Lösung (endlicher Zielfunktionswert),
2. L ist unlösbar, √
3. L ist unbeschränkt.

Lineare Programmierung

Fundamentalsatz der linearen Programmierung (2)

3. L ist unbeschränkt, wenn

- die Nebenbedingungen den Umfang nicht ausreichend einschränken und „Simplex“ die Eingangsvariablen beliebig erhöhen kann.

Lineare Programmierung

Fundamentalsatz der linearen Programmierung

Für jedes lineare Programm L, dass in Standardform gegeben ist, gilt genau eine der folgenden drei Aussagen:

1. L besitzt eine optimale Lösung (endlicher Zielfunktionswert),
2. L ist unlösbar, √
3. L ist unbeschränkt. √

Lineare Programmierung

Das Prinzip der Dualität

Beispiel:

Eine Großbrauerei braut zwei Sorten Bier.

1000 l Biersorte 1 : 1t Hopfen und 3t Malz

1000 l Biersorte 2 : 2t Hopfen und 1t Malz

1000 l Bier der Sorte 1 bringen 4 Geld Gewinn.

1000 l Bier der Sorte 2 bringen 6 Geld Gewinn.

Biersorte	Benötigter Hopfen	Benötigter Malz	Gewinn
1	1	3	6
2	2	1	4

Vorräte: 8 Tonnen Hopfen und 9 Tonnen Malz.

Lineare Programmierung

Das Prinzip der Dualität (2)

Problem: Rohstoffe verderblich!

Brauereibesitzer möchte Rohstoffe versichern:

y_1 : Versicherungswert für 1 Tonne Malz

y_2 : Versicherungswert für 1 Tonne Hopfen

$8\ y_1 + 9\ y_2$: Versicherungswert aller Vorräte (minimieren)

Um den Gewinn abzusichern, muss also mind. für

Biersorte 1 $\quad y_1 + 3\ y_2 >= 6$

Biersorte 2 $\quad 2\ y_1 + y_2 >= 4$

eine Versicherung abgeschlossen werden.

Lineare Programmierung

Das Prinzip der Dualität (3)

Problem 2: Einkommen maximieren!

Brauereimeister möchte wissen, wieviel Bier von beiden Sorten gebraut werden soll.

x_1 : Anzahl 1000 l Biersorte 1

x_2 : Anzahl 1000 l Biersorte 2

$6\ x_1 + 4\ x_2$: Verkaufswert der Biersorte

Um das Einkommen zu maximieren, kann also maximal für

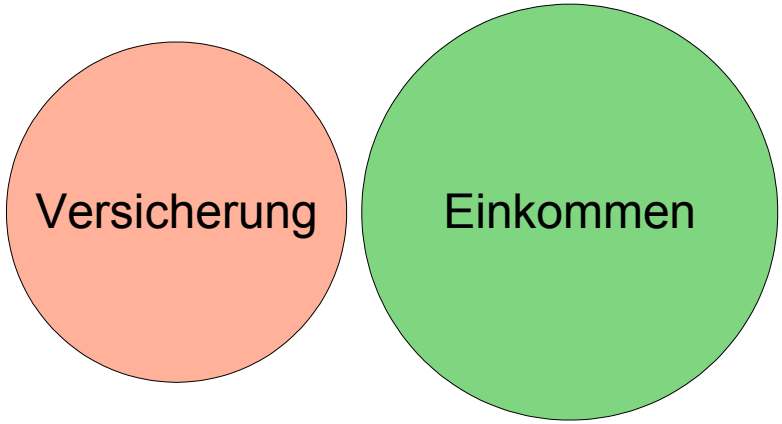
Hopfen	x_1	$+ 2\ x_2$	$<= 8$
Malz	$3\ x_1$	$+\ x_2$	$<= 9$

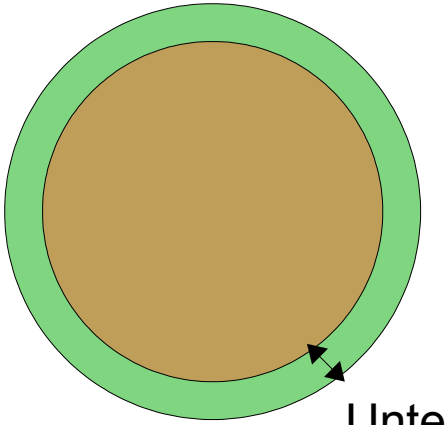
gebraut werden.

Lineare Programmierung

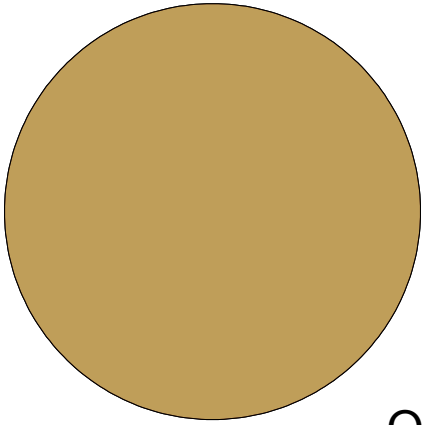
Das Prinzip der Dualität (4)

Problem 1 und 2 beschreiben eine Dualität!





Unterversichert



Optimum

Lineare Programmierung

Das Prinzip der Dualität (5)

(1) $\sum_{j=1}^{n} c_j x_j$ „Primales Programm“

(2)
$$\begin{aligned} \sum_{j=1}^{n} a_{ij} x_j &\leq b_i & i &= 1, 2, \cdots, m \\ x_j &\geq 0 & j &= 1, 2, \cdots, n \end{aligned}$$

(3) $\sum_{i=1}^{m} b_i y_i$ „Duales Programm“

(4)
$$\begin{aligned} \sum_{i=1}^{m} a_{ij} y_j &\geq c_j & j &= 1, 2, \cdots, n \\ y_i &\geq 0 & i &= 1, 2, \cdots, m \end{aligned}$$

Lineare Programmierung

Schwache Dualität

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i$$

Lineare Programmierung

Schwache Dualität

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i$$

nach (4)
$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_j\right) x_j$$
$$= \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j\right) y_j$$

nach (2)
$$\leq \sum_{i=1}^{m} b_i y_i$$

Lineare Programmierung

Schwache Dualität (2)

Sei x bzw. y eine zulässige Lösung für das primale Linearprogramm bzw das duale LP.

Haben beide den gleichen Zielfunktionswert, so ist dieser Wert optimal.

Wenn gilt $\sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i.$ dann optimal.

Beweis: Nach dem Prinzip der schwachen Dualität kann der Zielfunktionswert des primalen LP nicht den Wert des dualen LP übersteigen. Wenn also x und y den gleichen Zielfunktionswert haben, kann keine mehr verbessert werden.

Lineare Programmierung

Dualität der lineare Programmierung

Ausserdem gilt:

$$\sum_{j=1}^{n} c_j \bar{x}_j = \sum_{i=1}^{m} b_i \bar{y}_i$$

Ist x die optimale Lösung des primalen linearen Programms und y die optimale Lösung des dualen, dann haben beide immer den gleichen Zielfunktionswert.

Lineare Programmierung

Dualität der lineare Programmierung (2)

Beweis recht lang und komplex.

Idee:

- mittels Simplexalgorithmus finale Schlupfform erzeugen

$$z = v' + \sum_{j \in N} c'_j x_j$$

- aus Äquivalenz aller Schlupfformen für bel. Mengen von Werten haben die erste und letzte Schlupfform an einer Stelle den selben Wert.

$$\sum_{j=1}^{n} c_j x_j = v' + \sum_{j=1}^{n+m} c'_j x_j$$

Lineare Programmierung

Dualität der lineare Programmierung (3)

Idee (2):

$$\sum_{j=1}^{n} c_j x_j = v' + \sum_{j=1}^{n+m} c'_j x_j$$

- mittels Umformungen gelangt man zu

$$\sum_{j=1}^{n} c_j x_j = \left(v' - \sum_{i=1}^{m} b_j \bar{y}_j\right) + \sum_{j=1}^{n} \left(c'_j + \sum_{i=1}^{m} a_{ij} \bar{y}_i\right) x_j$$

Lineare Programmierung

Dualität der lineare Programmierung (4)

Idee (3):

$$\sum_{j=1}^{n} c_j x_j = (v' - \sum_{i=1}^{m} b_j \bar{y}_j) + \sum_{j=1}^{n} (c'_j + \sum_{i=1}^{m} a_{ij} \bar{y}_i) x_j$$

Dann folgt aufgrund dieses Lemma

$$Wenn \sum_{i \in I} a_i x_i = r + \sum_{i \in I} b_i x_i$$
$$dann\ gilt\ r = 0\ und\ a_i = b_i\ für\ alle\ i \in I$$

$$c'_j + \sum_{i=1}^{m} a_{ij} \bar{y}_i = c_j \qquad v' - \sum_{i=1}^{m} b_i \bar{y}_i = 0$$

Lineare Programmierung

Fundamentalsatz der linearen Programmierung

Für jedes lineare Programm L, dass in Standardform gegeben ist, gilt genau eine der folgenden drei Aussagen:

1. L besitzt eine optimale Lösung (endlicher Zielfunktionswert), √
2. L ist unlösbar, √
3. L ist unbeschränkt. √

=> Wenn INITIALIZE-SIMPLEX zulässige Lösung zurückgibt und SIMPLEX nicht mit unbeschränkt terminiert, dann ist die Lösung optimal.

Lineare Programmierung

Ende

Fragen, Anregungen und Beschimpfungen?

Lineare Programmierung

Quellen

- Cormen, Thomas; Leiserson, Charles;Rivest, Ronald; Stein, Clifford (2001): Introduction to Algorithms, Second Edition.
- Schlipf, Lena; Jankovic, Benjamin (2005): Lineare Programmierung Teil 1, http://page.mi.fu-berlin.de/~alt/vorlesungen/sem05/folien3.pdf
- Jiangsheng, Yu (2003): Linear Programming. http://icl.pku.edu.cn/member/yujs/papers/pdf/ComAlg9.pdf
- Riegler, Peter (2001): Dualität. Ergänzungen zum Kurs Operations Research, http://people.freenet.de/riegler/dualtext.pdf
- Wisniewski, Timothy (1996): The Simplex Java Applet, http://www-fp.mcs.anl.gov/otc/Guide/CaseStudies/simplex/applet/SimplexTool.html