Prof. Dr. Heinz Schmitz
Fachbereich Informatik
Hochschule Trier

Diskrete Mathematik, **Optimierung**, Statistik
Wintersemester 2018/2019
11. Januar 2019

Inhalt

Motivation, Organisatorisches.

1. Beispiele

   Beispiel *Möbelfabrik*, Software-Umgebung Gurobi/Python, Beispiel *Ölraffinerie*, **HA1.1**, Beispiel *Aufgabenverteilung*, Beispiel *Netzwerküberwachung*, Grundbegriffe: Lineare Funktion $f(x_1, \ldots, x_x) = \sum_{j=1}^{n} a_j x_j$ mit $a_j, x_j \in \mathbb{R}$, lineare (Un-) Gleichung, lineare Bedingungen, lineares Programm (LP), Schreibweise, nochmal *Möbelfabrik*, zulässige Lösungen, Zielfunktionswert $z = \sum c_j x_j$, optimale Lösungen, zulässiges LP, unbeschränktes LP, **HA1.2**.

2. Standard- und Slackform

   2.1. Standardform

   Definition max-LP mit n Variablen, m $\leq$-Bedingungen und n NN-Bedingungen. Äquivalenz linearer Programme, Umwandlung beliebiger LPs in äquivalente Standardform: Minimierung/Maximierung, NN-Bedingungen, $=$-Bedingungen, $\geq$-Bedingungen.

   2.2. Slackform

   Slackvariable, Struktur-/Entscheidungsvariablen, Konvertierung von Standardform-LP L in initiale Slackform L_S, Äquivalenz von L und L_S, Schreibweise $S : x_{n+1} = b_i - \sum_{j=1}^{n} a_{ij} x_j$ und $z = v + \sum_{j=1}^{n} c_j x_j$, **HA2.1**, Übersicht, Slackform von L, Basisvariablen, Basis, Indexmenge B, Nichtbasisvariablen, Indexmenge N, je zwei Slackformen von L mit gleicher Basis sind identisch, Basislösung von L, zulässige Basislösung mit NN-Bedingungen, zulässige Basis, Slackformen mit zulässiger Basis heißen zulässig, L_S ist zulässig gdw. $b_i \geq 0$ für alle i, **HA2.2**.

3. Das Simplex-Verfahren (SV) in Grundform

   Übersicht: Schrittweise Verbesserung durch wiederholte Iterationen auf zulässigen Slackformen.

3.1. Beispiel Iterationsschritt
Idee, drei Teilschritte: Auswahl einer NB-Variable (beitretende Variable), Auswahl einer B-Variable (verlassende Variable), Aufstellen der neuen Slackform (Pivotschritt), Pivotzeile, SIMPLEX, **HA3.1**.

3.2. Iterationen und Optimalität
Zulässiges S von L mit (B, N) gegeben, reduzierte Kosten, Schritt 1: Auswahl x_e mit $c_e > 0$, Korrektheit; Schritt 2: Auswahl x_l mit $l \in \{i \in B \mid a_{ie} > 0\}$ so dass b_l/a_{le} minimal, Korrektheit; Schritt 3: Neue zulässige Slackform S' von L mit (B', N'); degenerierte Basislösungen, Entstehung, degenerierte Iterationen, Iteration ist degeneriert gdw. z gleich bleibt; **HA3.2**.

3.3. Terminierung
Zyklus, Satz: SIMPLEX terminiert genau dann nicht, wenn ein Zyklus auftritt; Regel von Bland (Methode des kleinsten Index), **HA3.3**, Satz: SIMPLEX terminiert, falls stets die Regel von Bland angewendet wird; Zusammenfassung: Algorithmus SIMPLEX(S), Satz: Korrektheit von SIMPLEX(S) für jede zulässige Slackform S eines LP L in Standardform; Übersicht.

3.4. Eindeutigkeit optimaler Lösungen
Gilt in optimaler Basislösung $c_j < 0$ für alle $j \in N$, ist dies die einzige optimale Lösung für L; Ableitung linearer Bedingungen zur Beschreibung aller optimalen Lösungen aus der Slackform einer optimalen Basislösung, **HA3.4**.

3.5. Initialisierung
Hilfsproblem $H(L)$ für L mit unzulässigem L_S, LP L ist zulässig gdw. $H(L)$ optimale Lösung mit $x_0 = 0$ besitzt; $H(L)_S$ ist unzulässig, aber erste Iteration mit beitretendem x_0 liefert zulässige Slackform S_H, $H(L)$ ist beschränkt und zulässig, Ableitung einer zulässiger Slackform S_{init} von L aus optimaler Slackform S_H^{opt} von $H(L)$, Algorithmus INIT(L), Satz: Korrektheit von INIT(L), Übersicht, **HA3.5**.

3.6. Hauptsatz der Linearen Programmierung
Algorithmus SV (2-Phasen-Simplexverfahren), Hauptsatz, Bedeutung, Anmerkungen zur Laufzeit von SV, Interpretation der optimalen Slackform: nochmal *Möbelfabrik*, reduzierte Kosten der Strukturvariablen und der Slackvariablen (Schattenpreise).

4. LP-Dualität

Untere (primale) und obere Schranken.

4.1. Beispiel
Herleitung oberer Schranken durch Linearkombinationen der primalen Bedingungen.

4.2. Der Dualitätssatz

Duales Problem $D(L)$, duale Slackvariablen y_{m+j}, Korrespondenzen zwischen L und $D(L)$, Lemma über schwache Dualität, **HA4.1**, abgeleitetes Optimalitätskriterium $\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*$, Dualitätssatz, Folgerung über die Ermittlung optimaler Duallösungen y_i^* aus S_{opt}, duales Problem von $D(L)$ ist L, **HA4.2**, Zusammenhang zwischen Lösbarkeit von L und $D(L)$.

4.3. Satz vom komplementären Schlupf

Satz: Simultanes Optimalitätskriterium für L und $D(L)$, Folgerung: Rekonstruktion von $y_1^*, \ldots y_m^*$ aus $x_1^*, \ldots x_n^*$, **HA4.3**, Anwendung, **HA4.4**, Nichtdegenerierte Basislösung ist hinreichende Bedingung für die Rekonstruktion, Interpretation der Dualisierung in ökonomischen Modellen: nochmal *Möbelfabrik*.

5. Das Revidierte Simplex-Verfahren (RSV)

REVSIMPLEX(S) macht SV(L) zu RSV(S).

5.1. Slackformen in Matrix-Vektor-Schreibweise

Notationen für Standardform und Gleichungsform, initiale Slackform L_S ist $x_B = b - A_N x_N, z = 0 + c_N x_N$, Beispiel, falls A_B^{-1} existiert, ist die zugehörige Slackform $x_B = A_B^{-1} b - A_B^{-1} A_N x_N, z = c_B A_B^{-1} b + (c_N - c_B A_B^{-1} A_N) x_N$, Beispiel, ineffiziente Implementierung von SIMPLEX(S) mit Invertierung, Slackform mit Basis B existiert genau dann, wenn A_B regulär, **HA5.1**.

5.2. Beispiel Iterationsschritt

Schritt 1: $yA_B = c_B$, Schritt 2: $yA_N \overset{?}{<} c_N$, Schritt 3: $A_B d = a$, Schritt 4: $max_{t \geq 0}\ x_B^* - td \geq 0$, Schritt 5: $B', x_{B'}^*$; **HA5.2**.

5.3. Verfahren

Algorithmus REVSIMPLEX(S), Interpretation der Schirtte 1 und 2 als Anwendung des Satzes vom komplementären Schlupf, ökonomische Interpretation einer REVSIMPLEX-Iteration (temporäre Schattenpreise), Pricing.

6. Geometrie

Hyperebenen, Halbräume, Polyeder, Polytope, Menge der zulässigen LP-Lösungen sind Polytope, Korrespondenz von Seitenflächen und Variablen der Gleichungsform; geometrische Interpretation des Simpex-Verfahrens.

Literatur

[1] Vasek Chvatal. Linear programming. *A Series of Books in the Mathematical Sciences*, 1983.

[2] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. 2001.

[3] Jiri Matousek and Bernd Gärtner. *Understanding and Using Linear Programming*. Universitext. Springer Science & Business Media, Berlin, Heidelberg, July 2007.

[4] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to Linear Optimization*. January 1997.

[5] Albrecht Beutelspacher. *Lineare Algebra*. Eine Einführung in die Wissenschaft der Vektoren, Abbildungen und Matrizen. Springer-Verlag, Wiesbaden, November 2013.

[6] Winfried Hochstättler. *Lineare Optimierung*. Springer-Verlag, June 2017.

[7] Ravindra K Ahuja, Thomas L Magnanti, Sloan School of Management, and James B Orlin. *Network flows*. Nabu Press, September 2011.