Kapitel 2
Vorlesung zu Aktuellen Themen der Angewandten Mathematik

Vorlesung *Mathematik für Wirtschaftswissenschaftler* vom 31. Mai 2011

Stephan Dempe
Institut für Numerische Mathematik und Optimierung
TU Bergakademie Freiberg

Inhalt

Aufgabenstellung der Linearen Optimierung

$$\begin{aligned} c^\top x &\rightarrow \min \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

Simplexalgorithmus

Wichtige Formeln:

$$x = (x_B, x_N)\ x_N = 0,\ x_B = B^{-1}b$$

Basislösung; zulässig, falls $B^{-1}b \geq 0$.

$$\Delta_j = ((c_B^\top B^{-1}N)^\top - c_N)_j$$

Optimalitätsindikatoren

Wird eine Nichtbasisvariable x_{j_0} vergrößert, die anderen bleiben null:

$$(x_B =)B^{-1}b - B^{-1}A_j x_{j_0} \geq 0$$

wegen Zulässigkeit. Maximaler Wert von x_j:

$$x_j = \min\{\theta_i : (B^{-1}A)_{ij_0} > 0\} \text{ mit } \theta_i = \frac{(B^{-1}b)_i}{(B^{-1}A)_{ij_0}}$$

Die Simplextabelle

			c_1	$\dots$	c_m	c_{m+1}	$\dots$	c_n		
Nr.	x_B	c_B	x_1	$\dots$	x_m	x_{m+1}	$\dots$	x_n	b	θ
1	x_1	c_1	1	$\dots$	0	$\bar{a}_{1,m+1}$	$\dots$	$\bar{a}_{1n}$	$\bar{b}_1$	θ_1
2	x_2	c_2	0	$\dots$	0	$\bar{a}_{2,m+1}$	$\dots$	$\bar{a}_{2n}$	$\bar{b}_2$	θ_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
m	x_m	c_m	0	$\dots$	1	$\bar{a}_{m,m+1}$	$\dots$	$\bar{a}_{mn}$	$\bar{b}_m$	θ_m
			Δ_1	$\dots$	Δ_m	Δ_{m+1}	$\dots$	Δ_n	$\langle c_B, x_B \rangle$	

- Oben wurde angenommen, dass die ersten m Variable Basisvariable sind. Basisvariable können auch andere Variable sein.
- Der Wert der Basisvariablen in einer Basislösung steht in der Zeile der Basisvariablen in der Spalte b.
- Nicht unter den Basisvariablen vorkommende Variable sind Nichtbasisvariable. Diese haben in der Lösung den Wert Null.
- Der aktuelle Zielfunktionswert ist $\langle c_B, x_B \rangle = c_B^\top B^{-1} b$.

Simplextabelle in Kurzform

		$c^\top$		
BV	c_B	$x^\top$	b	θ
x_B	c_B	$B^{-1}A$	$B^{-1}b$	θ_i
		$\Delta^\top = c_B^\top B^{-1}A - c^\top$	$c_B^\top B^{-1}b$	

Der Simplexalgorithmus

1. Aufstellen der Simplextabelle
2. Wenn alle $\Delta_j \geq 0$ sind, dann ist die vorliegende Basislösung optimal.
   Sonst wähle ein j_0 mit $\Delta_{j_0} < 0$.
3. $(B^{-1}A)_{ij_0} \leq 0 \; \forall \, i \Longrightarrow$ (NF) nicht lösbar, stopp. Sonst bestimme
   $$\theta_{i_0} = \min\{\theta_i : (B^{-1}A)_{ij_0} > 0\}.$$
4. Umrechnung der gesamten Simplextabelle: Ersetze Basisvariable und Zielfunktionskoeffizienten in Spalten $x_B, \; c_B$. Schaffung einer Einheitsspalte mit einer Eins in der Zeile i_0 in der Spalte j_0 mit Hilfe des Gauß-Algorithmus.

Komplexität linearer Optimierungsaufgaben

Tabelle: Unterschied polynomialer zu exponentieller Rechenzeit

n	20	40	60	80
n^2	$4 \cdot 10^{-7}$ Sek.	$2 \cdot 10^{-6}$ Sek.	$4 \cdot 10^{-6}$ Sek.	$6 \cdot 10^{-6}$ Sek.
2^n	$1 \cdot 10^{-3}$ Sek.	18 Minuten	37 Jahre	$38 \cdot 10^6$ Jahre

Betrachtet sei die Aufgabe

$$\begin{array}{rcrcccccrcrcr}
- & 2^n x_1 & - & 2^{n-2} x_2 & - & \ldots & - & 2x_{n-1} & - & x_n & \to & \min \\
 & x_1 & & & & & & & & & \leq & 5 \\
 & 4x_1 & + & x_2 & & & & & & & \leq & 25 \\
 & 8x_1 & + & 4x_2 & + & x_3 & & & & & \leq & 125 \\
 & \vdots & & & & & \vdots & & & & \vdots & \vdots \\
 & 2^n x_1 & + & 2^{n-1} x_2 & + & \ldots & + & 4x_{n-1} & + & x_n & \leq & 5^n \\
 & & & & & & & & & x & \geq & 0.
\end{array}$$

Die (eindeutige) Optimale Lösung in diesem Beispiel ist $x^* = (0, 0, \ldots, 0, 5^n)$ mit dem optimalen Zielfunktionswert $z^* =^n$.

4 Variable:

Iteration	**Zielfunktionswert**	aufg. BV	ausg. BV
1	0		
2	40	x_1	x_1'
3	60	x_2	x_2'
4	100	x_1'	x_1
5	150	x_3	x_3'
6	190	x_1	x_1'
7	210	x_2'	x_2
8	250	x_1'	x_1
9	375	x_4	x_4'
10	415	x_1	x_1'
11	435	x_2	x_2'
12	475	x_1'	x_1
13	525	x_3'	x_3
14	565	x_1	x_1'
15	585	x_2'	x_2
16	625	x_1'	x_1

Newton-Algorithmus zur Lösung nichtlinearer Gleichungen

Löse $f(x) = 0$ mit einer differenzierbaren Funktion f.

Newton-Algorithmus zur Berechnung einer Lösung von $f(x) = 0$

Wähle x^0 hinreichend nahe an der zu berechnenden Nullstelle.

Für $k = 1, 2, \ldots$

Berechne $x^k = x^{k-1} + \frac{f(x^{k-1})}{f'(x^{k-1})}$.

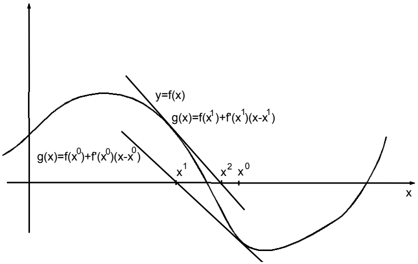

Abbildung: Tangente an den Graphen einer Funktion und Newton-Iteration

$f(x) = x^3 - 4x^2 - 7x + 10$
Es ist leicht einzusehen, dass $x = 1, x = 5$ und $x = -2$ Nullstellen dieser Funktion sind.
Wir beginnen mit $x^0 = -3$. Insgesamt ergibt sich der folgende Verfahrensablauf:

x^k	$f(x^k)$	$f'(x^k)$	x^{k+1}
-3	-32	44	-2.272
-2.272	$-6,4913$	$26,667$	$-2,029$
$-2,029$	$-0,626$	$21,59$	$-2,00040$
$-2,00040$	$-0,0084$	$21,008$	$-2,000000078$

Es sei die Funktion $f : \mathbb{R} \to \mathbb{R}$ zweimal stetig differenzierbar, $f(\overline{x}) = 0$ und $f'(\overline{x}) \neq 0$. Dann gibt es eine offene Umgebung U des Punktes $\overline{x}$, so dass das Newton-Verfahren für jeden Startwert $x^0 \in U$ quadratisch gegen die Nullstelle $\overline{x}$ konvergiert.

Dabei bedeutet die quadratische Konvergenz, dass es eine Konstante $K < \infty$ und einen Index k_0 gibt mit

$$|x^{k+1} - \overline{x}| \leq K|x^k - \overline{x}|^2 \; \forall \, k \geq k_0.$$

Lösung nichtlinearer Gleichungssysteme

Newton-Algorithmus zur Lösung von $f(x) = 0$ mit $f : \mathbb{R}^k \to \mathbb{R}^k$:

Wähle x^0 hinreichend nahe an der zu berechnenden Nullstelle.
Für $k = 1, 2, \ldots$
berechne x^k als Lösung des linearen Gleichungssystems

$$f(x^{k-1}) + Df(x^{k-1})(x - x^{k-1}) = 0. \tag{1}$$

Wenn $f : \mathbb{R}^k \to \mathbb{R}^k$ eine dreimal stetig differenzierbare Funktion ist, die an der Nullstelle $\widehat{x}$ der Funktion $f(x) = 0$ eine reguläre Jacobi-Matrix $Df(\widehat{x})$ besitzt, so ist das Newton-Verfahren in einer Umgebung U der Nullstelle $\widehat{x}$ anwendbar und konvergiert bei einem Start von einem Punkt $x^0 \in U$ quadratisch gegen die Nullstelle $\widehat{x}$.

Anwendung des Newton-Algorithmus zur Lösung linearer Optimierungsaufgaben

$$
\begin{aligned}
Ax &= b, \\
A^\top y + s &= c \\
x &\geq 0 \qquad\qquad (2) \\
s &\geq 0 \\
x^\top s &= 0.
\end{aligned}
$$

Lasse die Ungleichungen weg und erhalte ein nichtlineares Gleichungssystem:

$$
\begin{aligned}
Ax &= b, \\
A^\top y + s &= c \qquad\qquad (3) \\
x^\top s &= 0.
\end{aligned}
$$

Um den Newton-Algorithmus zur Lösung dieses Systems anwenden zu können, definieren wir uns die Funktion $F : \mathbb{R}^{n+m+n} \to \mathbb{R}^{m+n+n}$:

$$F(x, y, s) := \begin{pmatrix} Ax - b \\ A^\top y + s - c \\ Xs \end{pmatrix} \tag{4}$$

und lösen die Gleichung $F(x, y, s) = 0$ mit dem Newton-Algorithmus.

Dabei ist

$$\nabla F(x, y, s) = \begin{pmatrix} A & 0 & 0 \\ 0 & A^\top & E \\ S & 0 & X \end{pmatrix}$$

die Jacobi-Matrix der vektorwertigen Funktion F und E bezeichnet die Einheitsmatrix mit n Zeilen und n Spalten.

Diese Matrix ist quadratisch und regulär, wenn $x_i > 0,\ i = 1, \ldots, n$, $s_i > 0,\ i = 1, \ldots, n$ und die Matrix A linear unabhängige Zeilen besitzt.

Deshalb wird das Gleichungssystem $F(x, y, s) = 0$ regularisiert (gestört) und für $\mu > 0$ das nichtlineare Gleichungssystem

$$\begin{aligned} Ax &= b, \\ A^\top y + s &= c \\ Xs &= \mu e \end{aligned} \tag{5}$$

betrachtet, wobei $e = (1, \ldots, 1)^\top \in \mathbb{R}^n$ der n-dimensionale summierende Vektor ist.

Zu bemerken ist, dass

$$n\mu = \sum_{i=1}^{n} x_i s_i = c^\top x - b^\top y$$

Damit das System (5) lösbar ist, muss zumindest ein Punkt $(\widehat{x}, \widehat{y})$ existieren, für den

$$\widehat{x} \in \mathcal{X} := \{x : Ax = b,\ x_i > 0,\ i = 1, \ldots, n\}$$

und

$$\widehat{y} \in \mathcal{Y} := \{y : (A^\top y - c)_i < 0,\ i = 1, \ldots, n\}$$

gelten, da sonst $x_i(c - A^\top y)_i = \mu > 0,\ i = 1, \ldots, n$ nicht erfüllbar ist.

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^\top & E \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -r := -Xs + \mu e \end{pmatrix} \qquad (6)$$

Kurz-Schritt-Algorithmus zur Lösung linearer Optimierungsaufgaben:
Es seien $x^0 > 0,\ y^0,\ s^0 > 0$ mit $Ax^0 = b,\ A^\top y^0 + s^0 = c,\ X^0 s^0 - \mu_0 e = r^0$ sowie $\|r^0\|/\mu_0 \le 1/2$ gegeben. Wähle eine Abbruchgenauigkeit $\varepsilon > 0$ und setze $k := 0$.

Step 1: Bestimme eine Lösung $(\Delta x^k, \Delta y^k, \Delta s^k)^\top$ des linearen Gleichungssystems (6) mit $r = r^k$ und setze $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$.

Step 2: Verkleinere $\mu_{k+1} := \mu_k(1 - \frac{1}{6\sqrt{n}})$, berechne r^{k+1} und setze $k := k + 1$.

Step 3: Falls $\mu_k \le \varepsilon/n$, so beende den Algorithmus, sonst gehe zu Schritt 1.

Der Kurz-Schritt-Algorithmus endet nach spätestens

$$6\sqrt{n}\ln\frac{n\mu_0}{\varepsilon}$$

Iterationen mit einer Lösung x, y, s, für die
$x > 0,\ s > 0,\ Ax = b,\ A^\top y + s = c$ sowie

$$c^\top x - b^\top y \leq 2\varepsilon$$

gilt.