# Einführung in die Optimierung
# 7. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

**Fachbereich Mathematik** **WS 2013/14**
**Prof. Dr. Marc Pfetsch** **5./6.12.2013**
Dipl.-Math. Oec. Andreas Tillmann

**Rechnerübung / Programmieraufgaben**

Die Rechnerübungen können in Gruppen von bis zu 3 Personen bearbeitet werden. Zu Bearbeitung sind 8 Wochen vorgesehen; die Abgabe der Programme (per Email) muss bis zum 30.01.2014, 24 Uhr, an Ihren jeweiligen Übungsleiter **und** Andreas Tillmann erfolgen, Lösungen der zugehörigen Aufgaben sind in Ihren jeweiligen Übungen abzugeben (wie reguläre Hausübungen).

| Übungsgruppe | Abgabe per email an |
|---|---|
| 1 | palisa91@hotmail.com **und** tillmann@mathematik.tu-darmstadt.de |
| 2 | k.janzen90@web.de **und** tillmann@mathematik.tu-darmstadt.de |
| 3 | oli.habeck@googlemail.com **und** tillmann@mathematik.tu-darmstadt.de |
| 4 | radubogdan2@gmail.com **und** tillmann@mathematik.tu-darmstadt.de |
| 5 | thomas.kugler91@gmx.de **und** tillmann@mathematik.tu-darmstadt.de |
| 6 | k.janzen90@web.de **und** tillmann@mathematik.tu-darmstadt.de |

An den Tagen 5./6.12.2013 finden anstelle der normalen Übungen Rechnerübungen statt.

| Übungsgruppe | Tag | Zeit | Raum |
|---|---|---|---|
| 1 | Do., 5.12.2013 | 9.50 - 11.30 Uhr | S2\|15 K313 |
| 2 | Do., 5.12.2013 | 14.25 - 16.05 Uhr | S2\|15 K313 |
| 3 | Do., 5.12.2013 | 16.15 - 17.55 Uhr | S2\|15 K313 |
| 4 | Do., 5.12.2013 | 16.15 - 17.55 Uhr | S2\|15 K344 |
| 5 | Fr., 6.12.2013 | 9.50 - 11.30 Uhr | S2\|15 K344 |
| 6 | Do., 5.12.2013 | 14.25 - 16.05 Uhr | S2\|15 K344 |

Auf der EVS-Seite der Einführung in die Optimierung steht (wie dieses Blatt selbst unter Übungen) das `zip`-Archiv `LPtest.zip` mit Testdaten zu Verfügung, die zur Bearbeitung der Aufgabe R4 benötigt werden.

Ziel der Programmieraufgaben ist es, in MATLAB einen Löser für LPs der Form

$$\max\ c^T x \quad \text{s.t.} \quad Ax \le b,\ x \ge 0 \tag{P}$$

zu schreiben, der auf dem dualen Simplex-Verfahren und Erkenntnissen aus der Sensitivitätsanalyse beruht.

*Hinweis:* Berücksichtigen Sie bei Ihrer Implementierung das mögliche Auftreten von Rundungsfehlern! (Hierzu bietet es sich an, bei fraglichen Abfragen eine Toleranz von z.B. $10^{-6}$ anzusetzen.)

**Aufgabe R1** (Dualer Simplex-Algorithmus) (15 Punkte)
Implementieren Sie in MATLAB den dualen Simplex-Algorithmus aus der Vorlesung (Algorithmus 5.21) mit folgenden Auswahlregeln:

- Pricing: Dantzig-Regel ("Wähle ein $i \in \{1,\dots,m\}$ mit minimalem Wert $\bar{x}_{B_i} < 0$"; bei Uneindeutigkeit soll aus den Wahlmöglichkeiten das $i$ mit kleinstem $B_i$ gewählt werden)
- Ratio-Test: kleinster Variablen-Index ("Wähle das kleinste $j \in N$ mit $\frac{\bar{z}_j}{\alpha_j} = \min\left\{\frac{\bar{z}_k}{\alpha_k} : \alpha_k > 0,\ k \in N\right\}$")

Definieren Sie dazu die Funktion

```
[xOpt, B, message] = dualsimplex(A, b, c, initB)
```
Die Eingabe- und Ausgabe-Variablen sollen die folgende Bedeutung haben:

- Die Matrix `A` und die Vektoren `b` und `c` entsprechen hier denen aus der *Standardform*

$$\min\ c^T x \quad \text{s.t.} \quad Ax = b,\ x \geq 0.$$

  Dabei soll für $A \in \mathbb{R}^{m \times n}$ die Voraussetzung $\operatorname{rang}(A) = m$ gelten (also insbesondere auch $m \leq n$).
- Der Vektor `initB` soll eine dual zulässige Startbasis repräsentieren: Die $m$ Einträge des Vektors sollen die in der Basis enthaltenen Indizes sein.
- Der Vektor `xOpt` ist eine Optimallösung, falls eine solche existiert.
- Der Vektor `B` soll die berechnete optimale Basis enthalten (falls existent).
- Die Zeichenkette (*string*) `message` soll die Mitteilung "optimal" oder "unzulässig" beinhalten, bzw. "rang(A)<m" oder "Startbasis dual unzulässig", falls das Programm wegen Verletzung der jeweiligen Voraussetzung vorzeitig terminiert.

**Aufgabe R2** (Lösungsupdate bei Hinzunahme einer Ungleichung) (15 Punkte)
Implementieren Sie in MATLAB eine Funktion, die für ein gegebenes LP (P) mit bekannter (bezüglich Standardform) optimaler Basis nach Hinzunahme einer weiteren Restriktion der Form $\alpha^T x \leq \beta$ eine Nachbesserung der Lösung (Reoptimierung) mit dem dualen Simplex-Algorithmus aus Aufgabe R1 durchführt. Definieren Sie dazu die Funktion
```
[xOpt, B, message] = updatesol(A, b, c, oldB, newA, newb)
```
Die Eingabe- und Ausgabe-Variablen sollen hier folgende Bedeutung haben:

- Die Matrix `A` und die Vektor `b` und `c` entsprechen bei diesem Programm den Daten aus dem LP in der Form (P) – also *nicht* denen aus der Standardform!
- Der Vektor `oldB` soll die gegebene optimale Basis enthalten. (Beachten Sie: Diese Basis bezieht sich auf die zu (P) gehörende Standardform mit Schlupfvariablen.)
- Der Vektor `newA` und die Zahl `newb` entsprechen Koeffizientenvektor $\alpha$ bzw. rechter Seite $\beta$ der dem LP (P) hinzuzufügenden neuen Nebenbedingung $\alpha^T x \leq \beta$.
- Der Vektor `xOpt` ist eine Optimallösung des LPs

$$\max\ c^T x \quad \text{s.t.} \quad Ax \leq b,\ \alpha^T x \leq \beta,\ x \geq 0, \tag{P'}$$

  falls eine solche existiert.
- Der Vektor `B` soll (falls existent) die zugehörige optimale Basis enthalten (bzgl. der zu (P') gehörigen Standardform).
- Die Zeichenkette `message` soll die Mitteilung "optimal", "unzulässig" oder "rang(A)<m" beinhalten (analog zu Aufgabe R1).

**Aufgabe R3** (Inkrementeller Löser) (15 Punkte)
Implementieren Sie in MATLAB einen Löser, der ein in Form (P) gegebenes LP, für das keine (dual) zulässige Startbasis bekannt ist, durch wiederholtes Anwenden des dualen Simplex-Algorithmus löst. Definieren Sie dazu die Funktion
```
[xOpt, zOpt, B, message] = solvelp(A, b, c)
```
Die Eingabe- und Ausgabe-Variablen sollen folgende Bedeutung haben

- Die Matrix `A` und die Vektor `b` und `c` entsprechen den Daten aus dem LP in der Form (P).
- Der Vektor `xOpt` ist eine Optimallösung von (P), falls eine existiert.
- Die Zahl `zOpt` ist der entsprechende optimale Zielfunktionswert von (P).
- Der Vektor `B` repräsentiert die zugehörige optimale Basis (bzgl. der mit (P) assoziierten Standardform).
- Die Zeichenkette `message` soll die Mitteilung "optimal", "unzulässig", "unbeschränkt" oder "rang(A)<m" beinhalten.

Die Idee des zu implementierenden inkrementellen Lösungsverfahrens ist wie folgt:

- Da keine dual zulässige Startbasis bekannt ist, sollen die Ungleichungen des LPs (P) sukzessive hinzugefügt und die so erzeugten Teilprobleme jeweils mit dem dualen Simplex-Algorithmus reoptimiert werden.
- Es gilt insbesondere, sich zu überlegen, wie der erste Iterationsschritt dieses Vorgehens aussehen kann und wie sich die Update-Funktion aus Aufgabe R2 möglichst effizient ausnutzen lässt.
- Die Nichtnegativitätsrestriktionen sind immer (implizit) enthalten.

**Aufgabe R4** (Implementierungstests) (15 Punkte)

(a) Testen Sie Ihre Implementierung aus **Aufgabe R1** mit dem LP aus dem Beispiel 5.7 aus der Vorlesung (versuchen Sie die Startbasen $B = \{1,2,4\}$ und $B = \{1,2,5\}$) sowie mit dem folgenden LP (mögliche Startbasis: $B = \{3,4\}$):

$$\begin{array}{lrcrcl} \min & 2x_1 & + & x_2 & & \\ \text{s.t.} & x_1 & - & 2x_2 & \geq & 1 \\ & -x_1 & + & x_2 & \geq & 1 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

(b) Testen Sie Ihre Implementierung aus **Aufgabe R2** mit den folgenden Daten (angegeben bzgl. (P)):

i.
$$c^{(1)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad A^{(1)} = \begin{pmatrix} -1 & 1 \end{pmatrix}, \quad b^{(1)} = \begin{pmatrix} -1 \end{pmatrix}, \quad B^{(1)}_{opt} = \{1\},$$
neue Restriktion: $x_1 + x_2 \leq -1$.

ii.
$$c^{(2)} = \begin{pmatrix} -2 \\ -3 \\ -2 \\ -6 \end{pmatrix}, \quad A^{(2)} = \begin{pmatrix} -1 & 0 & -3 & 1 \\ -1 & -2 & 1 & -3 \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} -3 \\ 1 \end{pmatrix}, \quad B^{(2)}_{opt} = \{1,3\},$$
neue Restriktion: $-x_1 - x_2 + 2x_3 - 4x_4 \leq 3$.

iii.
$$c^{(3)} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \quad A^{(3)} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \quad b^{(3)} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad B^{(3)}_{opt} = \{1,2\},$$
neue Restriktion: $2x_1 + 9x_2 + 6x_3 \leq 2$.

iv.
$$c^{(4)} = c^{(3)}, \quad A^{(4)} = A^{(3)}, \quad b^{(4)} = b^{(3)}, \quad B^{(4)}_{opt} = \{2,5\},$$
neue Restriktion: $-x_1 + x_2 - 3x_3 \leq -1$.

(c) Geben Sie die Lösungen der im Folgenden angegebenen Optimierungsproblemen an! Verwenden Sie hierzu den von Ihnen entwickelten Code aus **Aufgabe R3**.

i. `LP1.mat`
ii. `LP2.mat`
iii. `LP3.mat`

(d) Erweitern Sie Ihre Programme jeweils um eine Ausgabe-Variable `iter`, die am Ende die Gesamtanzahl an Iterationen des dualen Simplex-Verfahrens beinhaltet, die zum Lösen eines gegebenen LPs benötigt wurden.

i. Wieviele Simplex-Iterationen vollzieht der inkrementelle Löser insgesamt, um das in `afiro.mat` gegebene LP zu lösen (falls möglich)?
ii. Erzeugen Sie mir der Funktion `kleeminty.m` Daten $A$, $b$ und $c$ eines LPs in Form (P) für die Eingaben `n=3`, `epsilon=1e-3` und `n=10`, `epsilon=1e-2`. Wieviele Iterationen werden jeweils zum Lösen dieser beiden LPs benötigt (falls möglich)? Wieviele Iterationen werden jeweils benötigt, wenn man den dualen Simplex mit den zugehörigen Startbasen aus `initBkl3.mat` bzw. `initBkl10.mat` direkt anwendet?

Geben Sie gegebenenfalls auch an, nach jeweils wievielen Iterationen und mit welcher Mitteilung Ihr Löser abbricht, wenn er nicht erfolgreich eine Optimallösung bestimmen kann.

Testen Sie Ihre Programme am besten auch mit weiteren Problemen (z.B. aus Skript und Hausübungen), um sicherzustellen, dass sie stabil laufen. Ihre Abgaben werden auch auf anderen als den hier angegebenen Instanzen getestet!

# Heute Mathe, morgen ???

***Mathematikerinnen erzählen***

**Vortragsreihe für Studierende der Mathematik**

jeweils Mittwoch, ab **14 Uhr** in **S1|01 A04**

| | | |
|---|---|---|
| **20. November** | *Dr. Katrin Schumacher* | Gestern Mathe, heute **Bosch** |
| **27. November** | *Prof. Dr. Christine Bach (h_da)* | Gestern Mathe, heute **Mathe** |
| **4. Dezember** | *Bianca Hinz* | Gestern Mathe, heute **Deutsche Bahn** |
| **11. Dezember** | *Laura Ströter* | Gestern Mathe, heute **Börse** |