Einführung in die Optimierung
7. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Prof. Dr. Winnifried Wollner
Alexander Matei, M.Sc.; Anna Walter, M.Sc.

WS 2020/21
17.12.2020

Rechnerübung

Die Rechnerübungen können in Gruppen von bis zu 4 Personen bearbeitet werden.

Das gesamte Aufgabenblatt zählt als Langzeit-Hausübung. Die **Abgabe** der Langzeit-Hausübung findet in der Übungsstunde am 21. Januar 2021 statt. Ihre Programme können Sie bis **Donnerstag, den 21. Januar 2021, 19:00 Uhr** in Moodle hochladen. Bitte beachten Sie, dass aus organisatorischen Gründen eine Abgabe nicht mehr geändert werden kann, sobald sie einmal hochgeladen wurde.

Für die *Programme* reicht *eine Abgabe pro Gruppe*. Bitte vermerken Sie die Namen aller Gruppenmitglieder als Kommentarzeile in jeder Funktion. Die *schriftliche Ausarbeitung* muss *von jedem Gruppenmitglied einzeln* abgegeben werden. Sie dürfen aber auch hier zusammenarbeiten, sollten Ihre Ergebnisse jedoch alleine aufschreiben. Geben Sie in diesem Fall ebenfalls die Namen aller Gruppenmitglieder an.

Denken Sie daran, Ihren Code ausreichend zu kommentieren, damit ggf. Teilpunkte vergeben werden können.

Hausübung

Aufgabe H1 (Primales Simplexverfahren) (20 Punkte)

Wir wollen in dieser Übung den Algorithmus 5.6 aus der Vorlesung in MATLAB implementieren. Dabei soll die Auswahlregeln von Dantzig benutzt werden. Es gibt dazu eine Vorlage in moodle, welche unter der Datei `SimplexDantzig.m` abliegt.

Diese enthält die Vorlage der Funktion

```
[xopt,B,message,iter, Zielfktnswert] = SimplexDantzig(A,b,c,Binit,xB).
```

Mit $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ die Daten eines Linearen Problems in Standardform

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Hierbei gilt $\operatorname{rang}(A) = m$ und der Vektor `Binit` enthält eine primal zulässige Startbasis.

Die Rückgabe besteht aus einer Optimallösung `xopt`, falls eine solche existiert, und der zugehörigen Basis `B`. Weiterhin wird eine Nachricht (*String*) `message` zurückgegeben, die angibt, ob ein Optimum gefunden wurde oder das LP unbeschränkt ist. Zuletzt wird die Anzahl `iter` der benötigten Iterationen sowie der Zielfunktionswert `Zielfktnswert` zurückgegeben.

Ihr Programm soll folgende Eingabefehler abfangen (`error('Fehlermeldung');`):

- inkompatible Matrixdimensionen von A, b, c
- A hat keinen vollen Zeilenrang
- `B` ist keine primal zulässige Startbasis

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 H2 (Testen) (7.5 Punkte)

Im moodle finden Sie ebenfalls folgende MATLAB Dateien:

- `Beispiel1.mat`

- `Beispiel2.mat`
- `Beispiel3.mat`
- `Beispiel4.mat`
- `Beispiel5.mat`

Testen Sie ihre Implementierung aus der Aufgabe H1 zunächst an diesen Beispielen und beschreiben Sie Ihre Ergebnisse.

Wie erklären Sie das Verhalten bei der Anwendung Ihres Codes auf Beispiel 5? Wie kann dieses Verhalten vermieden werden?
Lassen Sie Ihre Programme am besten auch mit weiteren Problemen (z.B. aus Skript und Hausübungen) durchlaufen, um sicherzustellen, dass sie stabil sind. Ihre Abgaben werden auch auf anderen als den hier angegebenen Instanzen getestet!

Aufgabe H3 (Auswahlregel nach Bland) (15 Punkte)
Implementieren sie auf Basis ihres Codes aus Aufgabe H1 eine weitere Funktion

`[xopt,B,message,iter, Zielfktnswert] = SimplexBland(A,b,c,Binit,xB).`

Diese Funktion soll nun die Auswahl im Pricing und im Ratio Test nach der Regel aus Satz 5.11 a) treffen.

Aufgabe H4 (Testen) (7.5 Punkte)
Testen Sie ihre Implementierung aus der Aufgabe H3 ebenfalls an den obigen Beispielen und beschreiben Sie Ihre Ergebnisse.

Aufgabe H5 (Kreiseln) (10 Punkte)
Nun sollen Sie das Verhalten aus der Aufgabe H2 mit `Beispiel5.mat` anhand folgendem Beispiel noch einmal genauer betrachten:
Gegeben sei das LP

$$\begin{array}{lrcrcrcrcl} \min & -2x_1 & - & 3x_2 & + & x_3 & + & 12x_4 & & \\ \text{s.t.} & -2x_1 & - & 9x_2 & + & x_3 & + & 9x_4 & \leq & 0 \\ & \frac{1}{3}x_1 & + & x_2 & - & \frac{1}{3}x_3 & - & 2x_4 & \leq & 0 \\ & & & & & & x_1,\ldots, & x_4 & \geq & 0 \end{array} .$$

Zeigen Sie, dass der Simplex-Algorithmus bei diesem Beispiel kreiseln (*cycle*) kann. Benutzen Sie dazu ihre Funktion `[xopt,B,message,iter, Zielfktnswert] = SimplexDantzig(A,b,c,Binit,xB)` und geben in jeder Iteration die aktuelle Basis an.
Überlegen Sie sich, wie das Kreiseln umgangen werden kann!