Optimierung 1, SS 2017

Implementationsaufgabe Pfadverfolgungsverfahren

Implementieren Sie in MATLAB (oder Octave wenn MATLAB nicht möglich) die in der Vorlesung vorgestellte Pfadverfolgungsmethode.
Ausgegangen wird von einem primal-dualen Paar von linearen Programmen in der Form

$$\max\{c^t x : Ax + w = b, x, w \geq 0\}$$

bzw.

$$\min\{b^t y : A^t y - z = c, y, z \geq 0\}$$

Die Eingabedaten sind eine $m \times n$ Matrix $A \in \mathrm{I\!R}^{\mathrm{m}\times\mathrm{n}}$, der Zielfunktionsvektor $c \in \mathrm{I\!R}^{\mathrm{n}}$ und der rechte Seitevektor $b \in \mathrm{I\!R}^{\mathrm{m}}$.

Achtung:

(1) Bei Vektoren wird immer davon ausgegangen, dass diese als Spalenvektoren an die Funktion übergeben werden.

(2) Bitte verwenden Sie nicht die „Elementweise Multiplikation“ zur Multiplikation eines Spaltenvektors a mit einem Zeilenvektor b, also $a \in \mathrm{I\!R}^{\mathrm{n}\times 1}$ und $b \in \mathrm{I\!R}^{1\times\mathrm{n}}$. In der Matlab Version R2016b soll eine derartige Multiplikation funktionieren, diese Version steht uns aber nicht zum Testen zur Verfügung. Bei früheren Matlab Versionen, die wir verwenden, führt diese Art Multiplikation zu Fehlermeldungen.

Das Resultat soll eine Funktion sein der Form

```
[x w y z Fall] = interior(A,b,c,xinit,winit,yinit,zinit,r,delta,epsilon,M,maxit)
```

A, b und c wurden bereits oben erklärt. `xinit`, `winit`, `yinit` und `zinit` spielen die Rolle von Startlösungen (x^0, w^0) und (y^0, z^0) für das primale bzw. duale Problem, wobei $x^0 > 0$, $w^0 > 0$, $y^0 > 0$ und $z^0 > 0$ ((x^0, w^0) bzw. (y^0, z^0) müssen jedoch für die jeweiligen Probleme nicht zulässig sein).

$r \in (0, 1)$ ist der Faktor der Schrittlänge (z.B. $r = 0.9$). Die Schrittlänge θ ist wie in der Vorlesung definiert, d.h.

$$\theta = \min\left\{1, \frac{r}{\max_{1\leq i\leq m, 1\leq j\leq n}\{\frac{|\Delta x_j|}{x_j}, \frac{|\Delta w_i|}{w_i}, \frac{|\Delta y_i|}{y_i}, \frac{|\Delta z_j|}{z_j}\}}\right\}.$$

$\varepsilon > 0$ und $M > 0$ (hinreichend groß) werden für das Abbruchkriterium (siehe unten) benötigt. $\delta > 0$ (z.B. $\delta = 0.1$) wird für die Bestimmung von μ benötigt.

maxit is eine (große) Zahl, die die maximale Anzahl von durchzuführenden Iteationen darstellt. Das heißt, der Algorithmus soll nach höchstens *maxit* Iterationen terminieren falls der Algorithmus nicht bereits (aufgrund eines anderen Terminierungskriteriums) terminiert hat.

Im folgenden bezeichne (x, w, y, z) die aktuelle Lösung im k-ten Schritt. Der Pfad-Parameter μ soll zu Beginn der Iteration als $\mu = \delta\frac{x^t z + y^t w}{n+m}$ gewählt werden.

- Wenn $||x||_\infty \geq M$ soll abgebrochen werden (primales Problem vermutlich unbeschränkt). Es soll `Fall=1` gesetzt werden.
- Wenn $||y||_\infty \geq M$ soll abgebrochen werden (duales Problem vermutlich unbeschränkt). Es soll `Fall=2` gesetzt werden.
- Wenn $||\rho||_1 = ||b - Ax - w||_1 \leq \varepsilon$, $||\sigma||_1 = ||c - A^t y + z||_1 \leq \varepsilon$ und $x^t z + y^t w \leq \varepsilon$, dann soll abgebrochen werden (P und D besitzen endliche Optimallösung) und `Fall=0` gesetzt werden.

- Wenn die Anzahl von *maxit* Iterationen erreicht wurde, dann soll abgebrochen werden und `Fall=4` gesetzt werden.

Als (x, w, y, z) soll der jeweils zuletzt betrachtete Iterationspunkt zurückgeliefert werden.

Testen Sie Ihren Code für verschiedene lineare Programme (z.B. solche aus der Übungsaufgabensammlung). Achten Sie darauf die unterschiedlichen Fälle (keine zulässige Lösung, unbeschränkt, endliche Optimallösung) abzudecken.
Vergewissern Sie sich zB., dass Ihr Programm folgende drei linearen Programme $P1$, $P2$ und $P3$ richtig löst. $P1$ besitzt eine Optimallsung $(x_1, x_3, x_3, x_4) = (0, 0, 3, 0)$ mit Zielfunktionswert 39, $P2$ ist unzulässig und $P3$ ist unbeschränkt.

$$(P1) \qquad \begin{array}{lrcrcrcrcr}
\max & 6x_1 & & & + & 13x_3 & + & 9x_4 & & \\
\text{unter} & & & & & & & & & \\
 & 2x_1 & + & x_2 & + & x_3 & + & 3x_4 & \leq & 5 \\
 & x_1 & + & 3x_2 & + & x_3 & + & 2x_4 & \leq & 3 \\
 & & & & x_1\,, & x_2\,, & x_3\,, & x_4 & \geq & 0\,.
\end{array}$$

$$(P2) \qquad \begin{array}{lrcrcr}
\max & x_1 & + & 3x_2 & & \\
\text{unter} & & & & & \\
 & -x_1 & - & x_2 & \leq & -3 \\
 & -x_1 & + & x_2 & \leq & -1 \\
 & 4x_1 & + & x_2 & \leq & 4 \\
 & & x_1\,, & x_2 & \geq & 0\,.
\end{array}$$

$$(P3) \qquad \begin{array}{lrcrcrcr}
\max & 3x_1 & + & x_2 & - & 4x_3 & & \\
\text{unter} & & & & & & & \\
 & x_1 & + & x_2 & - & 4x_3 & \leq & -1 \\
 & x_1 & - & x_2 & - & x_3 & \leq & -3 \\
 & & x_1\,, & x_2 & & x_3 & \geq & 0\,.
\end{array}$$

Darüberhinaus sollen Sie ihr Code an (mindestens) 6 weitere, von Ihnen erzeugte Instanzen linearer Programme testen, je zwei für die drei Fälle: a) Optimallösung existiert, b) Problem unbeschränkt, und c) Problem unzulässig. Erstellen Sie ein m-File `test_interior.m`, das 9 Test-Instanzen erzeugt (die drei vorgegeben und die 6 von Ihnen definierten) und durch den Aufruf von `interior` (approximativ) löst bzw. Unbeschränktheit oder Unzulässigkeit feststellt. `test_interior.m` sollte abgegeben werden.

Achtung: Bitte achten Sie darauf, dass die Funktion `interior` in der abgegebenen Version keine Debug-Ausgaben auf der Konsole macht.

Abgabe: Die Abgabe hat via TeachCenter in Form einer ZIP-Datei zu erfolgen. Die ZIP-Datei sollte alle notwendigen Files enthalten, also im Speziellen auch ein File interior.m mit der obengenannten Funktion sowie ein File `test_interior.m` (siehe oben). Die Abgabefrist ist Dienstag, der 6.6.2017, 23:55.
Nennen Sie Ihre Zip-Datei `int+Nachname+1.Buchstabe des Vornamen.zip` (lauter Kleinbuchstaben) - also z.B. bei Lena Musterfrau würde der Dateiname `intmusterfraul.zip` lauten.